



**Claus Bürbaumer**

**Entwicklung einer Testautomatisation**  
für betriebswirtschaftliche Planungssoftware

**Diplomarbeit**

zur Erreichung des akademischen  
Grades „Diplomingenieur“

Studienrichtung:

**Softwareentwicklung und Wirtschaft**

**Technische Universität Graz**

Fakultät für Informatik

O.Univ. Prof. Dipl.-Ing. Dr.techn. Ulrich Bauer  
Institut für Betriebswirtschaftslehre und Betriebssoziologie

Graz, im Jänner 2011

## **Eidesstattliche Erklärung**

Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Graz, am 07.01.2011

---

Claus Bürbaumer

## **Abstract**

The impact of automatic test systems is getting more and more relevant for the software engineering process. Testing during the development or tests on the final product are fundamental criteria for a successful software progress. In some projects the testing and maintenance process requires more time and resources than the development process itself. Every deployment which supports the testing-mechanism is crucial for the quality of the software. The same task significance of testing software in the engineering process has the corporate planning in a business operation. Every company is depending on a flexible and target-orientated planning. Nowadays there is some supportive software which helps to guide a company through the disturbed sea of the economy. It's required that these helpful solutions calculate the right results and have no malfunctions. Otherwise companies have a peck of trouble to be successful and profitable. Therefore a well-founded economic knowledge and knowledge about the software engineering process is required to provide testing-mechanism for these planning solutions. In the available work a test automatism was designed and developed for the planning software Professional Planner of the WINTERHELLER Company.

Keywords: Corporate Planning, Business Operation, Planning Software, Automatic Test Systems, Software Engineering, Professional Planner, WINTERHELLER

## **Kurzfassung**

Das Testen ist ein wichtiger Bestandteil für das Entwickeln von Software. Das automatische Testen von Software gewinnt immer mehr an Bedeutung im Zuge des Software-Entwicklungsprozesses. In vielen Projekten verbrauchen die Testphase und die Wartung der Software wesentlich mehr Ressourcen als der Entwicklungsprozess. Jegliche Entwicklung, die das Testen unterstützt oder effizienter gestaltet, trägt zur Qualitätssteigerung der Software bei. Genauso wichtig wie ein intakter Testprozess in der Softwareentwicklung ist eine betriebswirtschaftliche Planung für eine Unternehmung. Unternehmen sind auf eine flexible und zielorientierte Planung angewiesen. Es existieren EDV-Programme, die es den Unternehmen erleichtern, zukünftige Ereignisse besser planen zu können. Diese Lösungen müssen die Richtigkeit der Ergebnisse bewerkstelligen und Fehlfunktionen ausschließen. Andererseits können Entscheidungen, die auf den falschen Ergebnissen beruhen, zu existenzbedrohenden Situationen führen. Um Softwaretests für diese EDV-Lösungen zu entwickeln, bedarf es Qualifikationen auf betriebswirtschaftlichen sowie technischen Wissensgebieten. Die vorliegende Arbeit befasst sich mit der Entwicklung einer automatischen Testsoftware für die betriebswirtschaftliche Planungssoftware Professional Planner der Firma WINTERHELLER.

Schlüsselwörter: Betriebswirtschaftliche Planungssoftware, Finanzplanung, Automatisches Testen, Softwareentwicklung, Professional Planner, WINTERHELLER

## **Danksagung**

An dieser Stelle möchte ich mich bei all jenen bedanken, die mir während meines Studiums hilfreich und unterstützend zur Seite gestanden sind und mir bei der Erstellung meiner Diplomarbeit geholfen haben.

Ein herzliches Dankeschön gilt meiner Familie, besonders meinen Eltern, die mich während meines gesamten Studiums nicht nur finanziell sondern auch seelisch unterstützt und motiviert haben.

Ein besonderer Dank gilt auch der Firma Winterheller und in spezieller Form Herrn Hochleitner und seinem Team ohne die diese Arbeit nicht möglich gewesen wäre.

Abschließend bedanke ich mich bei den Personen, die mir bei der Erarbeitung dieser Diplomarbeit geholfen haben. Mein Dank gilt Herrn Dr. Bernd Zunk sowie Herrn Dipl.-Ing. Markus Ringhofer für die Betreuung meiner Arbeit.

Graz, im Jänner 2011

Claus Bürbaumer

# Inhaltsverzeichnis

1. Einleitung	1
1.1. Problemstellung und Zielsetzung der Arbeit	2
1.2. Aufbau der Diplomarbeit	3
2. Der Kooperationspartner WINTERHELLER software GmbH	5
2.1. Angebotene Softwarelösungen	5
2.1.1. Professional Planner	6
2.1.2. Professional Treasury	7
2.1.3. Professional Consolidation	7
2.2. Eingesetzte Technologien	7
3. Unternehmensplanung und Budgetierung	10
3.1. Unternehmensplanung	10
3.1.1. Planungsbegriff	11
3.1.2. Prognosebegriff	11
3.2. Budgetierung	12
4. Planung und Budgetierung im Unternehmen	13
4.1. Zweck und Ziele von Planung	13
4.1.1. Wirkung von Zielen	14
4.1.2. Die Rolle der Planung im Unternehmen	15
4.2. Strategische Unternehmensplanung	16
4.3. Operative Unternehmensplanung	16
4.4. Unternehmensplanung in Krisenzeiten	17

4.5.	Grenzen der Planung	18
4.5.1.	Die Zukunft ist nicht vorhersehbar	19
4.5.2.	Kein ökonomisch rationales Entscheidungsverhalten von Mitarbeitern	19
4.5.3.	Unternehmensspezifische Grenzen	19
4.6.	Risiken der Planung	20
4.7.	Aufgaben und Probleme der Budgetierung	20
4.8.	Das Unternehmensbudget	21
4.8.1.	Teil 1: Das Leistungsbudget	22
4.8.2.	Teil 2: Der Finanzplan	22
4.8.3.	Teil 3: Die Planbilanz	23
5.	Testen von Software	24
5.1.	Grundlagen zum Testen von Software	24
5.1.1.	Definition des Fehlerbegriffes	25
5.1.2.	Was versteht man unter Testen von Software	26
5.1.3.	Warum soll man testen?	26
5.1.4.	Allgemeine Prinzipien des Softwaretestens	27
5.1.5.	Wechselwirkung von Testen und Qualität der Software	28
5.2.	Zum Testprozess im Allgemeinen	29
5.2.1.	Teilprozess: Testplanung	29
5.2.2.	Teilprozess: Testanalyse und Testdesign	29
5.2.3.	Teilprozess: Testrealisierung und Testdurchführung	30
5.2.4.	Teilprozess: Testauswertung und Berichte	30
5.3.	Grundsätzliche Ausprägungen von Tests	31
5.3.1.	Funktionale Tests	31
5.3.2.	Nicht funktionale Tests	32

5.3.2.1.	Volume Test	33
5.3.2.2.	Stress Test	33
5.4.	Testautomatisation	33
5.4.1.	Motivation für Testautomatisation	34
5.4.2.	Einsatz von Werkzeugen	35
5.4.3.	Einsatzgebiete automatisierter Softwaretests	35
6.	Der Software-Entwicklungsprozess	37
6.1.	Prozessmodelle in der Softwareentwicklung	37
6.2.	Das ausgewähltes Vorgehensmodell Extreme Programming	39
7.	Ist-Analyse – Ausgangssituation	41
7.1.	Organisation der Testabteilung	41
7.2.	Bestandteile des Testablaufs	42
7.3.	Charakteristika der Testdaten	44
8.	Entwicklung der Winterheller TestSuite	45
8.1.	Festlegung von Zielen	45
8.2.	Entwurf und Design der Testsoftware	46
8.3.	Die Framework Definition	49
8.4.	Realisierung und Überprüfung der Zwischenergebnisse	51
8.5.	Testen der Software	52
9.	TestszENARIO	55
9.1.	Fallbeispiel Evolution Production GmbH	55
9.2.	Erfolgsplanung	56
9.3.	Produktionsplanung	58

9.4. Finanzplanung und Planbilanz	59
9.5. Investitionen	59
9.6. Steuern	60
9.7. Referenzdaten	60
<b>10. Potential der Testsoftware</b>	<b>63</b>
10.1. Vorteile durch die Verwendung der Testsoftware	63
10.2. Risiken durch die Verwendung der Testsoftware	63
10.3. Anwendungspotential der Testsoftware	64
<b>11. Zusammenfassung und Resümee</b>	<b>65</b>
Abkürzungsverzeichnis	66
Abbildungs- und Tabellenverzeichnis	67
Literaturverzeichnis	68
<b>ANHANG</b>	<b>72</b>

# 1. Einleitung

Betriebswirtschaftliche Planung hat in jeder Organisation Einzug gehalten bzw. war bereits seit jeher ein Teil jeder Unternehmung. Ohne eine korrekte Unternehmensplanung und der damit verbundenen Budgetierung kann eine Unternehmung in der heutigen Zeit nicht mehr am Markt überleben. Risiken müssen abgeschätzt, Potentiale bewertet bzw. ausgeschöpft und die Zahlungsfähigkeit für die unmittelbare Zukunft muss sichergestellt werden.<sup>1</sup>

Kaum eine Organisation kommt bei der Bewältigung dieser Vorgänge ohne eine elektronische Datenverarbeitung (EDV) aus.<sup>2</sup> Der Funktionsumfang der Programme wird immer größer und die Bedienbarkeit dadurch komplexer. Ebenso birgt ein zunehmend umfangreicheres Programm Potential für mögliche Fehler im System in sich. Der reibungslose Ablauf einer Unternehmung oder einer Organisation ist weitgehend von der Zuverlässigkeit eines Softwaresystems abhängig. Die Qualität der eingesetzten Software ist ein entscheidender Faktor für den Erfolg des Unternehmens geworden. Stellt Software in einem Unternehmen nicht nur Mittel zum Zweck, sondern den eigentlichen Unternehmensgegenstand dar, wird Fehlerfreiheit der Programme zum wichtigsten Qualitätskriterium. Langfristig kann Qualität nur durch einen gut organisierten Software-Entwicklungsprozess, sowie einen strukturierten Testmechanismus gewährleistet werden. Die Arbeitskraft der Mitarbeiter stellt eine wichtige Ressource in einem Betrieb dar. Der Einsatz dieser Ressource muss gezielt und effizient erfolgen um den Erfolg des Unternehmens gewährleisten zu können. Daher wurden in der Vergangenheit immer mehr Maschinen dort eingesetzt, wo menschliche Arbeitskraft eingespart werden konnte, um diese an anderen Stellen im Betrieb effektiver einsetzen zu können. Auch das Testen von Software bietet ein Potential zur Automatisierung durch sogenannte Testtools. Diese Programme ermöglichen die Automatisierungen einzelner Testvorgänge.<sup>3</sup>

---

<sup>1</sup> Vgl. GLEICH, R.; Klein, A. (2009), S. 27

<sup>2</sup> Vgl. GLEICH, R.; Klein, A. (2009), S. 43

<sup>3</sup> Vgl. SNEED, H.; BAUMGARTNER, M.; SEIDL, R. (2007), S. 181f.

## 1.1. Problemstellung und Zielsetzung der Arbeit

In den meisten Unternehmen ist eine händisch aufgebaute Unternehmens- bzw. Liquiditätsplanung aufgrund der Komplexität nicht mehr möglich. In diesen Fällen kommt eine professionelle Planungssoftware zur Anwendung, welche die unterschiedlichen Daten aus Buchführung und Kostenrechnung verarbeiten kann.<sup>4</sup>

Der, in dieser Arbeit untersuchten betriebswirtschaftlichen Planungssoftware, liegt eine komplexe Rechenlogik zugrunde. Mit dieser werden grundlegende rechnerische und unternehmerische Zusammenhänge innerhalb eines Betriebes festgelegt. Diese Art der Software funktioniert durch ein Zusammenspiel vieler betriebswirtschaftlicher und mathematischer Beziehungen. Die Fehlerfreiheit dieser Konstrukte ist Grundvoraussetzung für eine korrekt umgesetzte Planungssoftware.

Das regelmäßige Testen von Software trägt zu einer Qualitätssteigerung der selbigen bei.<sup>5</sup> Automatisierte Testwerkzeuge reduzieren den Aufwand und Zeitrahmen nicht sofort, Investitionen dahingehend zahlen sich jedoch meist nach einiger Zeit aus. Werden automatisierte Tests für einen längeren Zeitraum ausgeführt, verkürzen diese die Testzeit, erhöhen die Zuverlässigkeit der zu testenden Software und entlasten mitunter das gesamte Testteam.<sup>6</sup>

Das Testen nimmt einen wesentlichen Teil des Softwareentwicklungsprozesses ein. Aus diesem Grund stellt sich die Frage nach der Art und Intensität der Tests. Nur eine effizient organisierte Testabteilung kann gewünschte Ergebnisse in einer vorgegebenen Zeit liefern.

Das Ziel dieser Diplomarbeit ist es, für die betriebswirtschaftliche Planungssoftware des Kooperationspartners WINTERHELLER software GmbH, ein Testtool für das automatische Testen der Softwarelösung zu erstellen. Das Testtool soll die Funktionsweise des fertigen Programms testen. Dazu ist es nötig ein Testszenario zu entwerfen, welches aus

---

<sup>4</sup> Vgl. ERICHSEN, J. (2010); S. 177f.

<sup>5</sup> Vgl. FRÜHAUF, K.; LUDEWIG, J.; SANDMAYR, H. (2006), S. 20

<sup>6</sup> Vgl. DUSTIN, E.; RASHKA, J.; PAUL, J. (2000); S. 37

Eingabedaten und erwarteten Ausgabedaten besteht. Anhand dieses Szenarios werden die betriebswirtschaftlichen und mathematischen Beziehungen auf deren Richtigkeit geprüft. Die Unternehmensplanung wird ebenso Untersuchungsgegenstand dieser Arbeit sein wie der Entwurf eines geeigneten Testszenarios. Dafür müssen theoretische Grundlagen vom betriebswirtschaftlichen Planungswesen erarbeitet werden. Die Diplomarbeit besteht somit aus einem betriebswirtschaftlichen und einem technisch motivierten Teil.

## **1.2. Aufbau der Diplomarbeit**

Die vorliegende Diplomarbeit gliedert sich in 11 Kapitel, welche sich aus mehreren Unterkapiteln zusammensetzen. Die theoretischen Grundlagen werden in den ersten sechs Kapiteln erarbeitet. Die folgenden vier Kapitel behandeln den praktischen Teil der Diplomarbeit. Die Arbeit wird durch eine Zusammenfassung und ein Resümee beschlossen.

Die Problemstellung und Zielsetzung der vorliegenden Diplomarbeit wird in Kapitel 1 motiviert. Darin wird die Notwendigkeit betriebswirtschaftlicher Planungssoftware für Unternehmen und Softwaretests im Zuge der Entwicklung von Computerprogrammen unterstrichen.

Der praktische Teil der Diplomarbeit wurde mit einem Kooperationspartner, der Firma WINTERHELLER software GmbH, durchgeführt. Im 2. Kapitel werden das Unternehmen und dessen Softwarelösungen, sowie die eingesetzten Technologien kurz vorgestellt. Untersuchungsgegenstand der Diplomarbeit ist die angebotene Softwarelösung Professional Planner. Der Professional Planner der Firma WINTERHELLER software GmbH wird auf dem Gebiet der Unternehmensplanung eingesetzt. Was unter Unternehmensplanung und der damit verbundenen Budgetierung verstanden wird, erörtert Kapitel 3.

Kapitel 4 setzt sich mit der Thematik auseinander, warum Planung und Budgetierung in Unternehmen wichtig sind. Die Herausforderungen und Ziele werden ebenso besprochen wie auch die Grenzen der Planung sowie etwaige Risiken des Planungsprozesses.

Das 5. Kapitel widmet sich dem technischen Bereich der Diplomarbeit und befasst sich mit dem Testen von Software. Basismodelle werden aufgezeigt und grundlegende Mechanismen dargelegt. Die Automatisierung von Tests wird ebenfalls theoretisch aufbereitet.

Da es in diesem Projekt um die Erstellung einer Testsoftware für den Kooperationspartner WINTERHELLER software GmbH ging, wird in Kapitel 6 auf die wichtigsten Grundlagen des Software-Entwicklungsprozesses eingegangen. Das eingesetzte Prozessmodell wird erklärt und dessen Verwendung begründet.

Im Blickpunkt von Kapitel 7 steht die Ist-Analyse, welche auch gleichzeitig den Übergang von der Theorie- zum Praxisteil darstellt. Dabei werden die Organisation der Testabteilung, der Testablauf und die verwendeten Testdaten analysiert. Die gewonnenen Daten über die Art der Tests, sowie der Organisation des Testablaufs fließen in das Design der zu erstellenden Testsoftware ein.

Der Entwicklungsprozess der WINTERHELLER TestSuite steht im Mittelpunkt von Kapitel 8. Die Ideen zum Design des Programms werden erklärt und ein grober Überblick über das Framework der Testumgebung wird gegeben.

Für den Einsatz der Testsoftware wurde ein Testszenario konzipiert, welches in Kapitel 9 vorgestellt wird. Darin werden die Art der Eingangsdaten, sowie der dazugehörigen Ausgangs- oder Referenzdaten erklärt. Die vollständigen Angaben des Szenarios befinden sich im Anhang der Diplomarbeit.

Im Kapitel 10 werden das Anwendungspotential bzw. die Vorteile der entwickelten Testsoftware beleuchtet. Die, durch den Einsatz der vorgestellten Testsoftware, möglichen Risiken werden ebenfalls erörtert.

Eine Zusammenfassung über das gesamte Projekt sowie ein Resümee bilden in Kapitel 11 den Abschluss dieser Diplomarbeit.

## **2. Der Kooperationspartner WINTERHELLER software GmbH**

Das Unternehmen, mit der Hauptniederlassung in Graz (Österreich), wurde 1988 von Dr. Manfred Winterheller gegründet. Die WINTERHELLER software GmbH steht seit über 20 Jahren für höchste Flexibilität, Produktivität und Effektivität im Bereich von Softwarelösungen für die Unternehmenssteuerung von mittelständischen Unternehmen und Konzernen.

Als Softwarehersteller mit hoher Lösungskompetenz und Spezialist in den Bereichen Business Intelligence (BI) und Corporate Performance Management (CPM) ist die WINTERHELLER software GmbH der Ansprechpartner für Unternehmen aller Branchen. Die Firma WINTERHELLER software GmbH bietet fundiertes betriebswirtschaftliches Wissen, professionelle Beratung und Implementierung sowie flexible Softwarelösungen für die Unternehmensplanung, -steuerung und -konsolidierung. Als Microsoft Gold Certified Partner und Citrix Partner garantiert das Unternehmen modernste Technologie.<sup>7</sup>

### **2.1. Angebotene Softwarelösungen**

Die Firma WINTERHELLER software GmbH zeichnet sich durch Lösungen für Berater und Controllingfachleute, CPM-Lösungen für Fachabteilungen, strategische CPM-Lösungen für Konzernanforderungen und Branchen-Lösungen aus. Die Softwarepakete des Unternehmens ermitteln selbstständig Kennzahlen und berechnen sämtliche abhängige Werte. Die jeweils integrierte Logik ermöglicht das Abbilden der verschiedensten Geschäftsfälle. Da jeden Tag Entscheidungen getroffen werden müssen, sind Entscheider mit großen Verantwortungen betraut. Sie sind auf der Suche nach Hilfsmitteln, die ihnen diese verantwortungsvollen Tätigkeiten erleichtern. Mit den Softwarepaketen der Firma WINTERHELLER software GmbH können Unternehmen und deren Mitarbeiter in wenigen Augenblicken verschiedenste

---

<sup>7</sup> Vgl. WINTERHELLER software GmbH. (2010). Abgefragt 10.12.2010, von <http://www.winterheller.com>

Szenarien simulieren und vergleichen. Das gibt die Sicherheit Situationen richtig zu bewerten und Entscheidungen für morgen zu treffen.<sup>8</sup>

### 2.1.1. Professional Planner

Professional Planner (PP) ist die Budgetierungs- und Planungslösung für jede Unternehmensgröße, unabhängig von der Branche und dem Geschäftsmodell, und das schnell, einfach und zu 100 % transparent. Wie Abbildung 1 zeigt, ist es möglich, verschiedene Auswertungen und Vergleiche auf Basis betriebswirtschaftlicher Planungsrechnung zu erstellen.



Abbildung 1: Professional Planner<sup>9</sup>

Professional Planner garantiert durch seine einzigartige OLCAP-Technik eine konsistente Erfolgs-, Liquiditäts- und Bilanzrechnung. Gleichzeitig ermöglicht diese Technik beliebige Simulationsszenarien, wie sie im Planungsprozess immer wieder als 'Was-wäre-wenn'-Fragen beantwortet werden müssen.<sup>10</sup>

<sup>8</sup> Vgl. WINTERHELLER software GmbH. (2010). Abgefragt 10.12.2010, von <http://www.winterheller.com>

<sup>9</sup> WINTERHELLER software GmbH. (2010). Abgefragt 10.12.2010, von <http://www.winterheller.com>

<sup>10</sup> Vgl. WINTERHELLER software GmbH. (2010). Abgefragt 10.12.2010, von <http://www.winterheller.com>

### **2.1.2. Professional Treasury**

Professional Treasury unterstützt bei der Erstellung von Liquiditätsplänen und bringt Sicherheit, Steuerbarkeit und Effizienz in die Liquiditätsplanung. Das Programm verknüpft offene Postenlisten aus der Finanzbuchhaltung mit den Daten der Monats- und Jahresplanung. Durch die Verknüpfung dieser beiden Datenbestände wird gewährleistet, dass eine lückenlose und korrekte kurz- bis langfristige Vorhersage der Unternehmensliquidität erstellt werden kann.<sup>11</sup>

### **2.1.3. Professional Consolidation**

Diese Softwarelösung organisiert alle Prozesse, die zur legalen Konsolidierung und zur Managementkonsolidierung von Daten für das interne und externe Konzernberichtswesen benötigt werden. Mit Professional Consolidation ist es möglich effizient hochkomplexe Konsolidierungsprozesse zu steuern und zu überwachen. Das Tool verwaltet und konsolidiert klassische Kontensalden und Controlling-Dimensionen wie Kostenstellen, Kostenträger und Profitcenter.<sup>12</sup>

## **2.2. Eingesetzte Technologien**

Die Softwareprodukte sind skalierbare Lösungen, die in der klassischen 3-Schichten-Architektur mit Datenbank-Server, Applikations-Server und User Interface aufgebaut sind. Auf einem Datenbank-Server wird Microsoft SQL Server installiert. Dieser beinhaltet alle erforderlichen Datenbanken. Die Daten werden aufbereitet, transformiert und in ein Datawarehouse<sup>13</sup> geladen und stellen so den Datenpool für alle WINTERHELLER Produkte. Die Software von WINTERHELLER software GmbH wird auf einem Applikations-Server installiert.

---

<sup>11</sup> Vgl. WINTERHELLER software GmbH. (2010). Abgefragt 10.12.2010, von <http://www.winterheller.com>

<sup>12</sup> Vgl. WINTERHELLER software GmbH. (2010). Abgefragt 10.12.2010, von <http://www.winterheller.com>

<sup>13</sup> Datawarehouse = Zentrale Datensammlung, deren Inhalte sich aus Daten unterschiedlicher Quellen zusammensetzt.

Durch den Einsatz der OLCAP-Technologie stehen dem Anwender betriebswirtschaftliche Grundlagen und Zusammenhänge zur Verfügung, die von Beginn weg sofort nutzbar sind. Die OLCAP-Technologie stellt eine firmeneigene Weiterentwicklung der OLAP<sup>14</sup> Technologie dar. Wie in einer multidimensionalen Datenbank stehen auch hier Strukturelemente zur Verfügung. Diese Strukturelemente sind mit entsprechenden Eigenschaften für eine betriebswirtschaftliche Verwendung vorbelegt. Dadurch enthalten die Teilstücke (Strukturelemente) alle nötigen Informationen um sich zu einem integrierten Gesamtsystem zusammenzufügen. Der Vorteil liegt in einer hohen Flexibilität, da spezifische Anforderungen des Unternehmens als Eigenschaften einzelner Strukturelemente ohne Programmieraufwand erfasst werden können.

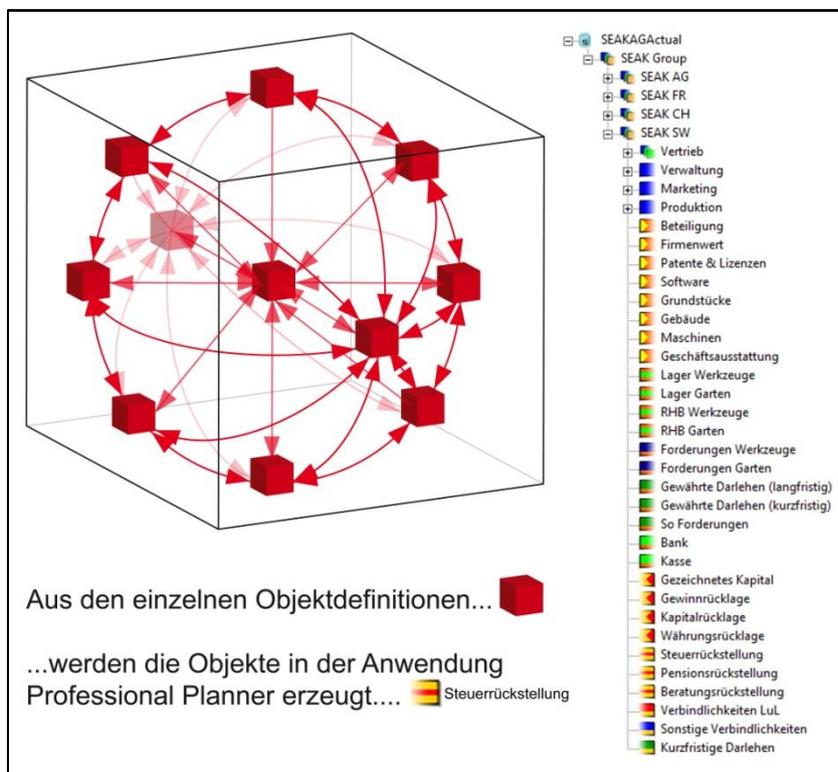


Abbildung 2: OLCAP-Würfel<sup>15</sup>

Diese Elemente werden über drei Dimensionen wie Abbildung 2 zeigt definiert. Ähnlich wie bei objektorientierten Programmiersprachen basiert OLCAP auf Objekten. Die Objektdefinitionen insgesamt bilden eine Business Content Library (BCL). Das bedeutet, dass es für jeden betriebswirtschaftlichen Begriff, den wir zur Beschreibung eines Business Modells benötigen (Kostenstelle, Bilanzkonto, Investitionsobjekt etc.), es eine

<sup>14</sup> OLAP = Online Analytical Processing

<sup>15</sup> WINTERHELLER software GmbH. (2010). Abgefragt 10.12.2010, von <http://www.winterheller.com/software/technologie.html>

Objektdefinition in der BCL gibt. Die Objektdefinitionen sehen verschiedene Eigenschaften vor, die in den jeweiligen Objekten unterschiedlich eingestellt werden können. Die Objektdefinitionen der BCL setzen sich wiederum aus vielen verschiedenen untergeordneten Objektdefinitionen zusammen. Diese untergeordneten Objektdefinitionen stehen für einzelne kleine Berechnungsschritte, Aggregations-Logiken, für Verknüpfungen mit anderen Objekten. Entscheidend für die Flexibilität dieses Konzeptes ist nicht unbedingt, wie viele verschiedene betriebswirtschaftliche Objektdefinitionen die BCL enthält. Vielmehr ist entscheidend, dass die BCL für konkrete Anwendungsfälle erweitert werden kann. Dafür werden aus verschiedenen untergeordneten Objektdefinitionen selbständig neue betriebswirtschaftliche Objektdefinitionen zusammengefügt. OLCAP ist eine Gesamtheit von untergeordneten Objektdefinitionen, die zu beliebigen betriebswirtschaftlichen Objektdefinitionen zusammengesetzt werden können. Diese untergeordneten Objekte enthalten sämtliche Logiken, die in betriebswirtschaftlichen Anwendungen benötigt werden.<sup>16</sup>

---

<sup>16</sup> whitepaper\_olcap.pdf, 10.12.2010

### 3. Unternehmensplanung und Budgetierung

In diesem Kapitel wird die Bedeutung von Unternehmensplanung und Budgetierung behandelt.

Planung beginnt mit der Festlegung von Zielen. Diese sind Definitionen von Vorstellungen bzw. von Wünschen und machen den Kern jeder Planung aus. Um Ziele der Unternehmung zu erkennen, Zielkonflikte erkennbar zu machen und die Ressourcenverteilung optimal vornehmen zu können, ist die Unternehmensplanung ein unverzichtbares Mittel der Unternehmensführung.<sup>17</sup>

Die Budgetierung wird als wichtiger Teil der Unternehmensplanung gesehen. Budgetierung kann als operative und wertmäßige Planung verstanden werden, die sich aktiv mit der Zukunft des Unternehmens und seiner Bereiche auseinandersetzt.<sup>18</sup>

#### 3.1. Unternehmensplanung

Ziele zeigen, was ein Unternehmen erreichen will, die Planung zeigt den Weg dorthin. Das Wort Planung stammt aus dem Lateinischen. Planus bedeutet flach, glatt, eben oder im übertragenen Sinne: reibungslos, deutlich, klar, durchschaubar. Planung bedeutet, künftige Gegebenheiten und Zusammenhänge so weit wie möglich überschaubar zu machen, um das eigene Entscheiden und Handeln auf ein gewolltes Ziel hin möglichst reibungslos gestalten zu können.<sup>19</sup>

In der Betriebswirtschaftslehre werden die Begriffe Prognose und Planung nicht immer getrennt voneinander betrachtet. Die Planung versucht die Zukunft aktiv zu gestalten und die Prognose bleibt stets eine passive Vorschau.<sup>20</sup>

---

<sup>17</sup> Vgl. KRETSCHMER, P. (1979), S. 14ff.

<sup>18</sup> Vgl. TREUZ, W. (2004), S. 340

<sup>19</sup> Vgl. EISL, C.; HANGL, C.; LOSBICHLER, H.; MAYR, A. (2008), S. 42

<sup>20</sup> Vgl. FISCHER, J. (1989), S. 20

Die folgenden Definitionen sollen für den weiteren Verlauf dieser Arbeit die beiden Begriffe inhaltlich voneinander abgrenzen.

### **3.1.1. Planungsbegriff**

Planen kann als zielorientiertes, zukunftsgerichtetes, verknüpfendes und folgerichtiges Denken, sowie – aus Gründen der Zweckmäßigkeit – als die Dokumentation dieses Vorganges, bezeichnet werden.<sup>21</sup>

Planung ist der Entwurf einer Ordnung, nach der sich das betriebliche Geschehen in der Zukunft vollziehen soll. Sie ist das gedankliche und systematische Gestalten des zukünftigen Handelns. Dabei basiert die Planung auf Analysen des Istzustandes und der Zukunft des Unternehmens und der Umwelt. Sie verarbeitet eine Vielzahl von Informationen über Fakten, Entwicklungen, Trends und Verhaltensweisen und trägt, bei Berücksichtigung der Ungewissheit hinsichtlich des Eintritts und der Realisierung von Annahmen, zur Kursfixierung bei.<sup>22</sup>

### **3.1.2. Prognosebegriff**

Prognosen sind deskriptive Aussagen mit hypothetischem Charakter über zukünftige Zustände und Ereignisse. Von diesen Aussagen wird gefordert, dass sie empirischen Informationsgehalt besitzen und an Hand einer hinreichenden qualifizierten Erfahrungsbasis empirisch begründet oder begründbar sind.<sup>23</sup>

Prognosen verbinden die in der Vergangenheit gemachten betrieblichen Erfahrungen sowie die vorhandenen wissenschaftlichen Erkenntnissen mit den prospektiven Überlegungen der Planung und Entscheidung.<sup>24</sup>

---

<sup>21</sup> Vgl. VOSSBEIN, R. (1974), S. 13

<sup>22</sup> Vgl. EHRMANN, H. (2007), S. 19

<sup>23</sup> Vgl. WILD, J. (1969), S. 60

<sup>24</sup> Vgl. WILD, J. (1969), S. 62

### 3.2. Budgetierung

Der Begriff Budget entstand aus dem französischen Wort bougette, das Lederbeutel oder Brieftasche bedeutet. Im letzten Jahrhundert hatte der britische Schatzkanzler die Angewohnheit den Steuerbedarf für das kommende Jahr mittels seiner offenen Brieftasche zu verdeutlichen. Um 1870 verband man mit dem Begriff Budget das Schriftstück, auf dem der Steuerbedarf ausgewiesen war.<sup>25</sup>

Die Budgetierung wurde Anfang des 20. Jahrhundert entwickelt und wird seither in einer Vielzahl von Unternehmen und Organisationen unterschiedlichster Branchen und Größen eingesetzt. Ihre weite Verbreitung und große Bedeutung führten zu zahlreichen Publikationen, in denen der aktuelle Stand der Budgetierung aufgezeigt wird oder Wissenschaftler und Praktiker Empfehlungen zur Ausgestaltung der Budgetierung ableiten.<sup>26</sup>

Nach HORVATH ist ein Budget ein formalzielorientierter, in wertmäßigen Größen formulierter Plan, der eine Entscheidungseinheit für eine bestimmte Zeitperiode mit einem bestimmten Verbindlichkeitsgrad vorgegeben wird.<sup>27</sup> HEISER setzt die Budgetierung der kurzfristigen Planung gleich wobei er die Planperiode maximal auf ein Jahr begrenzt (= operative Budgetierung).<sup>28</sup>

---

<sup>25</sup> Vgl. STEINER, G.A. (1969), S. 294

<sup>26</sup> Vgl. GLEICH, R.; Klein, A. (2009), S. 37

<sup>27</sup> Vgl. HORVATH, P. (1998), S. 224

<sup>28</sup> Vgl. HEISER, H.C. (1964), S. 35

## 4. Planung und Budgetierung im Unternehmen

Die Bedeutung von Plan und Ziel ist nicht nur in vorgegebene Formblätter einzugeben, welche Umsätze oder Kosten man für die kommende Periode erwartet, Planung ist ein kreativer Prozess, der diese Daten zum Ergebnis hat. Sie entstehen während eines Prozesses und bilden sein Ende. Sie sind nicht der eigentliche Planungszweck. Von der obersten Unternehmensspitze wird eine Marschrichtung vorgegeben. Dieser globale Plan und die damit verbundenen Ziele werden dann auf die hierarchisch dafür notwendigen Einheiten herunter gebrochen. Diese erarbeiten Maßnahmen, um die gewünschten Ergebnisse erreichen zu können. Diese Maßnahmen in Euro bewertet ergeben das dafür notwendige Budget.<sup>29</sup>

Im nächsten Unterkapitel werden die Ziele sowie der Zweck der Planung dargelegt. Dabei wird besonders auf die Wichtigkeit von Zielen eingegangen. Die Unterteilung der Unternehmensplanung in eine strategische und operative Planung, wird in Kapitel 4.2. und 4.3. diskutiert.

### 4.1. Zweck und Ziele von Planung

Planung bildet als systematischer, zukunftsorientierter Willensbildungsprozess den Kernbestandteil jeder Führung. Der Planung liegen stets Ziele zugrunde, die autonom festgelegt oder systematisch abgeleitet sein können.<sup>30</sup>

Es ist seit jeher unbestritten, dass Menschen mit Zielen über einen klareren Fokus und eine höhere Leistungsbereitschaft verfügen. Der Leistungssport ist dem Business in diesem Punkt sehr ähnlich. Sieht man sich herausragende Persönlichkeiten an wie zum Beispiel Lance Armstrong, Hermann Meier, Pete Sampras, Tiger Woods oder Michael Schumacher, so

---

<sup>29</sup> Vgl. STARK, P. (2006), S. 16

<sup>30</sup> Vgl. HAHN, D.; HUNGENBERG, H. (2001), S. 99

haben sie eines gemeinsam: den absoluten Willen zu siegen und das Ziel, der Beste in der jeweiligen Sportart zu sein. Diesem Ziel ordnen sie ihr gesamtes Leben unter.<sup>31</sup>

#### 4.1.1. Wirkung von Zielen

Wie wichtig es ist, sich Ziele zu setzen, zeigt das Experiment, das die Sportuniversität Wien in den siebziger Jahren durchgeführt hat. Eine Profi-Hochspringerin, die zu den Top-Athletinnen Europas zählte, sollte einen Sprung ohne die Hochsprunglatte machen. Sie sollte alles so machen wie immer, beim optimalen Punkt starten und den Radius genau so laufen wie im Wettkampf. Das überraschende Ergebnis: Ohne sichtbares Ziel erreichte die Profisportlerin nur 70 Prozent ihrer gewohnten Leistung. Sie blieb 30 Prozent unter ihren Möglichkeiten wie Abbildung 3 zeigt.<sup>32</sup>

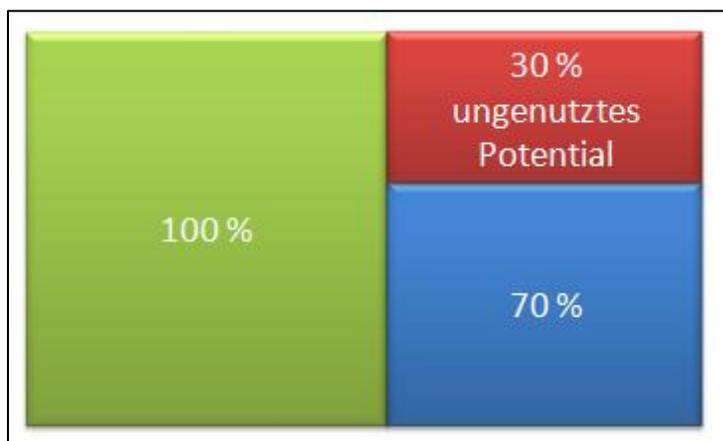


Abbildung 3: Ungenutztes Leistungspotenzial<sup>33</sup>

Das Beispiel zeigt, dass ca. 30 Prozent unseres Leistungspotential brach liegen, wenn wir keine Ziele haben. Es zeigt somit wie wichtig Ziele in unserem täglichen Leben und somit auch in einer Unternehmung sind.

---

<sup>31</sup> Vgl. STARK, P. (2006), S. 17ff.

<sup>32</sup> Vgl. STARK, P. (2006), S. 18

<sup>33</sup> Vgl. STARK, P. (2006), S. 18

#### 4.1.2. Die Rolle der Planung im Unternehmen

Die Planung als Führungsinstrument nimmt im Unternehmen eine bedeutende Stellung ein. Sie trägt dazu bei, wichtige Ziele zu erreichen und wichtige Maßnahmen zu ergreifen, Erwartungen und Einstellungen zu bilden und zu überprüfen, zu agieren und zu reagieren sowie innerbetriebliche Kontakte zu fördern. Die Planung kann als ein hervorragendes Mittel der Unternehmenspolitik angesehen werden. Im Einzelnen sprechen folgende Punkte für die bedeutende Rolle der Planung im Unternehmen:<sup>34</sup>

- „Die Planung trägt zum Erkennen und Strukturieren von Problemen des Gesamtunternehmens sowie der unterschiedlichsten Unternehmensbereiche bei.
- Die Planung zwingt zu wirtschaftlichem Denken und sachlichem Vorgehen.
- Die Planung veranlasst, Erwartungen und Einstellungen zu bilden und zu überprüfen, gelegentlich sogar in Frage zu stellen.
- Die Planung bewirkt, dass die damit befassten Mitarbeiter das Unternehmen nicht als die Summe einzelner Bereiche, sondern als Ganzes sehen.
- Die Planung führt zu einer Identifizierung von Mitarbeitern mit dem Unternehmen und wichtigen Aktionen.
- Die Planung veranlasst zum Denken in Zeitabschnitten, zu längerfristigem Denken und zum Erkennen von Horizonten.
- Die Planung trägt zu kreativem Denken, zum Entwickeln von Fantasie bei.
- Die Planung fördert problemorientierte und problemlösungsorientierte Vorgehensweisen.
- Die Planung hilft, Ziele zu entwickeln und zu variieren.
- Die Planung bereitet wichtige Entscheidungen vor, bis zu solchen, die die Existenz des Unternehmens betreffen.
- Die Planung motiviert Mitarbeiter, Ziele zu erreichen und wichtige Entscheidungen zu treffen.
- Die Planung fördert Kommunikation auf horizontaler und vertikaler Ebene.
- Die Planung koordiniert Ziele und Maßnahmen einzelner Bereiche.

---

<sup>34</sup> EHRMANN, H. (2007), S. 19

- Die Planung zwingt zum Überdenken von ad-hoc-Entscheidungen.
- Die Planung ermöglicht Soll-ist-Vergleiche und schafft damit Kontrollmöglichkeiten.
- Die Planung zwingt zu schnellem Reagieren auf neue Situationen.“

Die angeführten Punkte zeigen, dass Planung für zielgerichtetes Handeln unabdingbar ist. Planung dient auch zum Anreiz und zur Motivation, Koordination und Integration, Information und Dokumentation, Zielsetzung, Steuerung, Innovation und Zielerreichung.<sup>35</sup>

## **4.2. Strategische Unternehmensplanung**

Die strategische Unternehmensplanung befasst sich mit der Festlegung relativ unspezifischer, hochaggrierter und abstrakter qualitativer und quantitativer Merkmale. Die Festlegung der Produktparten für eine Unternehmung ist zum Beispiel qualitativer Natur, die Verteilung der Vertriebsressourcen auf die Sparten ist ein quantitatives Merkmal. Solche hochaggrierten Rahmenbedingungen der Unternehmensaktivität lassen sich in der Regel nur langsam verändern. Aus diesem Grund muss die strategische Planung langfristig angelegt sein und hat sich ebenso mit gegenwärtigen wie mit zukünftigen Tätigkeitsbereichen zu befassen.<sup>36</sup>

## **4.3. Operative Unternehmensplanung**

Die operative Planung hat die „Operationalisierung“ des strategischen Plans für die nähere Zukunft als Gegenstand. Unter Operationalisierung verstehen wir die Ausfüllung des relativ unspezifischen strategischen Planungsrahmens durch spezifische und konkrete Maßnahmen, die als Aktionsprogramme für die mittlere und untere Führungsebene geeignet und kommunizierbar sind. Der operative Plan stellt also keinen unabhängigen Plan, sondern die Konkretisierung der ersten Schritte des strategischen Plans dar. Diese auf Aktion zielende

---

<sup>35</sup> Vgl. FINK, C. (2003), S. 97

<sup>36</sup> Vgl. HANSSMANN, F. (1990), S. 3

Konkretisierung zu weit in die Zukunft zu treiben, hat angesichts der zu erwartenden Revision keinen Sinn. Der operative Plan kann Maßnahmen unterschiedlicher Detaillierung und Konkretisierung sowie unterschiedlicher zeitlicher Wirkungen enthalten.<sup>37</sup>

#### **4.4. Unternehmensplanung in Krisenzeiten**

Allgemeine Wirtschaftskrisen mit Konjunkturerinbrüchen und sinkende Kaufkraft, wie es in den letzten Jahren der Fall war, bedeuten nicht zwangsläufig den Ruin aller Unternehmen. Da Wirtschaftskrisen in der Regel zeitlich begrenzt sind, besteht für gut geführte Unternehmen auch in Wirtschaftskrisen eine Möglichkeiten, gestärkt aus der Krise hervorzugehen.<sup>38</sup>

Stabilität bedeutet in einem schwierigen Umfeld, sich auf den ursprünglichen Zweck der Planung zu besinnen. Die Zukunft möglichst genau zu prognostizieren und Budgets exakt einzuhalten, bedeutet nicht gleichzeitig für Krisenzeiten gewappnet zu sein. Die Unternehmung muss bestmöglich auf die Zukunft und mögliche Krisen vorbereitet sein. Dafür ist ein kritisches Durchdenken verschiedener Handlungsalternativen nötig, um wertvolle Früherkennungsinformationen zur Steuerung des Unternehmens zu erhalten.<sup>39</sup>

Im Hinblick auf die Prognose potenzieller Unternehmenskrisen ist die Erarbeitung verschiedener Handlungsalternativen von unschätzbarem Wert. Szenarien und dazugehörige Entscheidungsmöglichkeiten werden skizziert, Extremszenarien werden dargestellt und die entsprechenden Grenzen der Handlungsmöglichkeiten durchgespielt. Solche auch als Worst-Case- und Best-Case-Szenarien bezeichneten Zukunftsbilder können neben Chancen (Best-Case-Szenarium) auch Szenarien mit überlebenskritischen Bedrohungen als Worst-Case-

---

<sup>37</sup> Vgl. HANSSMANN, F. (1990), S. 4

<sup>38</sup> Vgl. BAIER, P. (2007), S. 420

<sup>39</sup> Vgl. GLEICH, R.; KLEIN, A. (2009), S. 28

Szenarien enthalten. Wie Abbildung 4 zeigt, werden diese Extremszenarien neben dem erwarteten Szenario in die Planung mit einbezogen.<sup>40</sup>

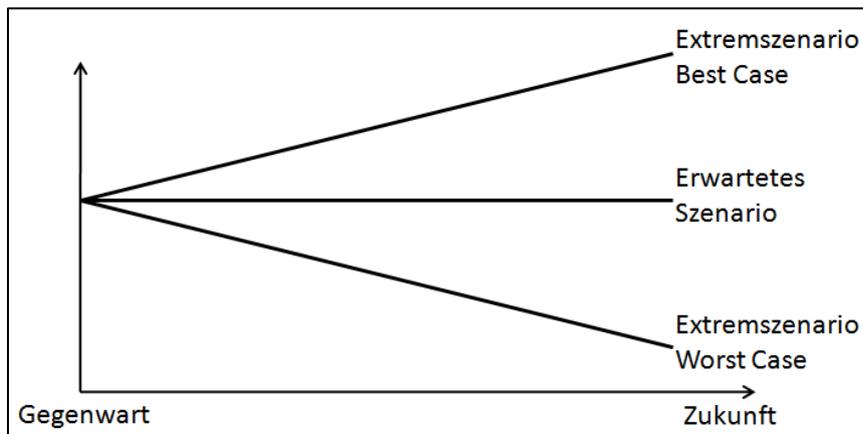


Abbildung 4: Szenarien<sup>41</sup>

Eine Unternehmensplanung kann auf Basis relevanter Analysen und Prognosen sowie unter Verwendung geeigneter Planungs-/Entscheidungstechniken viele künftige Unternehmenskrisen antizipieren. Durch eine zielorientierter Durchführung und Kontrolle der geplanten Strategien/Maßnahmen können Krisen verhindert werden. Die Unternehmensplanung kann jedoch dieser Funktion häufig nicht gerecht werden, aufgrund spezifischer Grenzen der Planung.<sup>42</sup>

#### 4.5. Grenzen der Planung

Dass in Plänen festgesteckte Ziele nicht immer erreicht werden, ist eine alltägliche Erfahrung. Auch große Unternehmen sind vor Fehlkalkulationen nicht sicher. Welche Ursachen könnten dafür verantwortlich sein? Einerseits ist die Zukunft nur schwer oder überhaupt nicht vorhersehbar, andererseits ist das Entscheidungsverhalten von Personen nicht immer ökonomisch und rational motiviert. Zusätzlich existieren unternehmensspezifische Grenzen, welche die Planung beeinflussen.

---

<sup>40</sup> Vgl. KRYSTEK, U.; MOLDENHAUER, R. (2007), S. 83f.

<sup>41</sup> Quelle: Eigene Darstellung

<sup>42</sup> Vgl. KRYSTEK, U.; MOLDENHAUER, R. (2007), S. 89

#### **4.5.1. Die Zukunft ist nicht vorhersehbar**

Jeder realistische Plan benötigt Prognoseinformationen. Dazu zählen Absatzchancen, Produktionszahlen oder ein vermutetes Verhalten von Mitarbeitern. Diese, für eine Prognose nötigen Informationen, lassen sich nicht immer erheben oder messen. Diese Unsicherheit stellt ein Problem für das Unternehmen dar, denn Erfolg kann in einem Wettbewerbsmarkt nur in einer Welt mit nicht vorhersehbarer Zukunft entstehen. Diese Unsicherheiten eröffnen Erfolgchancen in einem umkämpften Markt. Die Unklarheit darüber, welche Kunden bei welchem Unternehmen einkaufen, eröffnet Chancen neue Kunden zu sich zu ziehen.<sup>43</sup>

#### **4.5.2. Kein ökonomisch rationales Entscheidungsverhalten von Mitarbeitern**

Die klassische ökonomische Theorie geht davon aus, dass sich reale Menschen bei jeder Gelegenheit wie Nutzenmaximierer verhalten. Menschliches Verhalten ist jedoch sehr vielschichtig, vielgestaltig und daher nie wirklich vorhersehbar. Zum Beispiel sind Reaktionen von Außendienstmitarbeitern auf neue Zielvorgaben oder Prämiensysteme nicht wirklich planbar, man ist auf Schätzungen und Erfahrungswerte angewiesen.<sup>44</sup>

#### **4.5.3. Unternehmensspezifische Grenzen**

Eine strategische Planung allein mit Mitteln der Kostenrechnung aufzubauen, wäre nicht zielführend. Unpassende Methoden und Instrumente zur Planung zählen ebenso zu den unternehmensspezifischen Grenzen, wie eine unpassende Organisation der Planung und Kontrolle. Für eine erfolgreiche Planung sind unbedingt Erfahrung und Wissen notwendig. Beides betrifft sowohl den Planungsgegenstand (Absatz, Produktion) als auch die Organisation und den Ablauf einer Planung sowie die Planungsmethoden.<sup>45</sup>

---

<sup>43</sup> Vgl. RIEG, R. (2008), S. 34

<sup>44</sup> Vgl. RIEG, R. (2008), S. 35

<sup>45</sup> Vgl. RIEG, R. (2008), S. 35

Werden diese Grenzen der Planbarkeit nicht in Überlegungen mit einbezogen führen diese zu Risiken für die Unternehmung.

#### **4.6. Risiken der Planung**

Unternehmerische Tätigkeiten werden ebenso von Risiken begleitet wie von Chancen des Erfolgs. Risiken lassen sich als Möglichkeit oder Gefahr der Zielverfehlung bzw. Nichterreicherung von gesetzten Zielen interpretieren.<sup>46</sup>

Planung birgt gewisse Risiken, welche darin liegen können, dass

- „von nicht erreichbaren Zielen ausgegangen wird,
- unrealistische Annahmen gemacht werden,
- ‚Nicht-Planbares‘ geplant wird,
- nichtvorhersehbare Ereignisse und Entwicklungen zu falschen Planansätzen führen,
- Enttäuschungen, die durch Fehleinschätzungen entstehen, zu Frustration und mangelnder Bereitschaft für weitere Planungen führen,
- Planung in manchen Fällen mit hohem Aufwand verbunden ist.“<sup>47</sup>

Diese Risiken müssen bei jeder Planung in Betracht gezogen und weitgehend ausgeschlossen werden, damit die Umsetzung des Geplanten auch realistisch und machbar ist.

#### **4.7. Aufgaben und Probleme der Budgetierung**

Christian Schmahl, Segmentvorstand Finanzen & Controlling der Deutschen Telekom AG, beschreibt die Anforderungen an eine moderne Budgetierung folgendermaßen:

„Das Budget muss einfach und klar sein, damit jeder weiß, was es bedeutet. Es muss aber vor allem auch flexibel sein, gerade weil sich die Gegebenheiten schnell verändern können

---

<sup>46</sup> Vgl. KRYSSTEK, U.; MOLDENHAUER, R. (2007), S. 30

<sup>47</sup> EHRMANN, H. (2007), S. 20

und das Geschäft nicht eindeutig planbar ist, d.h., eine Budgetierung darf, nicht in Stein gemeißelt sein, sondern muss immer eine gewisse Flexibilität mitbringen. Natürlich wird auf oberster Ebene ein Rahmen festgelegt, der dann auch gültig ist, aber in der unterjährigen Ausgestaltung müssen Änderungen möglich sein. Um die Anforderungen zusammenzufassen: klar, einfach, zeitgerecht und flexibel.“<sup>48</sup>

Die traditionelle Budgetierung impliziert eine gute Vorherseh- und Abschätzbarkeit der zukünftigen Entwicklungen. In Zeiten von Krisen und Strukturbrüchen sind Budgets oft nicht mehr nutzbar bzw. müssen neu geplant werden. Die Steuerung des Budgets wird durch die zunehmende Dynamik und Komplexität der Unternehmensumwelt erschwert.<sup>49</sup>

#### 4.8. Das Unternehmensbudget

Das Budget und die Planung sind miteinander verknüpft, denn die Planung, strategisch oder operativ, basiert auf der Strategie, die durch das Management vorgegeben wird. Die Budgets wiederum leiten sich aus der Planung ab und detaillieren diese. Budgets sollen das Ergebnis bzw. die Auswirkungen der festgelegten Strategie darstellen und geben so dem Management Aufschluss über mögliche Probleme. Die einzelnen Budgets werden im Zuge eines Abstimmungsprozesses aufeinander abgestimmt.<sup>50</sup>

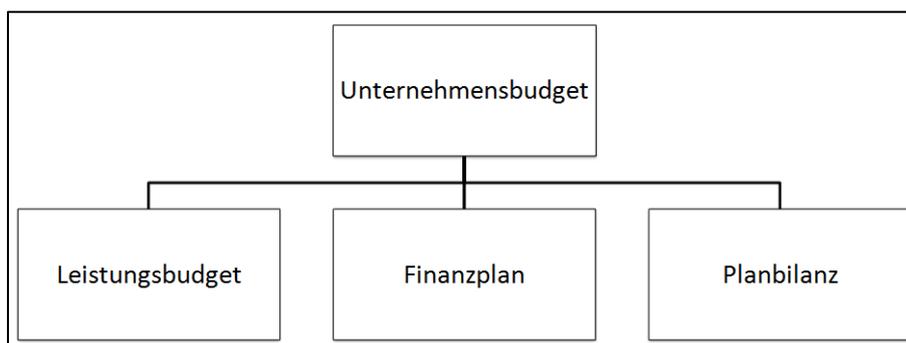


Abbildung 5.: Unternehmensbudget<sup>51</sup>

<sup>48</sup> GLEICH, R.; Klein, A. (2009), S. 18

<sup>49</sup> Vgl. GLEICH, R.; Klein, A. (2009), S. 65

<sup>50</sup> VGL. HORNGREN, C.T.; FOSTER, G.; DATAR, S.M. (2000), S. 179

<sup>51</sup> Quelle: Eigene Darstellung

Das Unternehmensbudget setzt sich, wie Abbildung 5 zeigt, aus den drei großen, miteinander verknüpften und voneinander gegenseitig abhängigen Teilplänen

- Leistungsbudget,
- Finanzplan und
- Planbilanz

zusammen.

#### 4.8.1. Teil 1: Das Leistungsbudget

Aufbauend auf die Vorgaben stellt das Leistungsbudget die Gewinn- und Verlustrechnung (GuV) für den Planungszeitraum dar.

	Geplante Erlöse	
-	geplante variable Kosten zu Standardwerten	
=	Deckungsbeitrag	
-	geplante Fixkosten	
=	Betriebsergebnis	
+/-	Betriebsüberleitung	
=	<b>Unternehmensergebnis auf Standardwertbasis</b>	

Tabelle 1: Unternehmensergebnis<sup>52</sup>

#### 4.8.2. Teil 2: Der Finanzplan

Im Zuge der Erstellung des Finanzplans werden die budgetierten Aufwendungen und Erträge in Einnahmen und Ausgaben transformiert. Die erfolgsneutralen Veränderungen in den Aktiven und Passiven sind ebenfalls zu erfassen. Dadurch ist es möglich den tatsächlichen Zahlungsmittelbedarf bzw. –überschuss der Planungsperiode vorherzubestimmen.

Das folgende Schema des Finanzplanes wurde aus dem Buch von EGGER/WINTERHELLER „Die kurzfristige Unternehmensplanung“ entnommen, da der untersuchten Softwarelösung Professional Planner dieses Buch als betriebswirtschaftliche Basis zu Grunde liegt.

---

<sup>52</sup> Quelle: Eigene Darstellung

#### **Grundsätzliches Schema des Finanzplanes:**

	Ergebnis nach Steuern gemäß Leistungsbudget
+	nicht ausgabewirksame Aufwendungen
-	nicht einnahmewirksame Erträge
=	Zahlungsmittelzu- bzw. -abfluss auf Grund des Leistungsbudgets (Cashflow)
<hr/>	
+	erfolgsneutrale Zahlungseingänge durch
	Herabsetzen der Aktiven
	Erhöhung der Passiven
	Einzahlungen durch den Inhaber oder die Gesellschafter
-	erfolgsneutrale Zahlungsausgänge durch
	Erhöhung der Aktiven
	Senkung der Passiven
	Auszahlung an den Inhaber oder die Gesellschafter
<hr/>	
=	<b>Zahlungsmittelbedarf bzw. -überschuss</b>

**Tabelle 2: Finanzplan**<sup>53</sup>

#### **4.8.3. Teil 3: Die Planbilanz**

Die Planbilanz ergibt sich zwingend aus Leistungsbudget und Finanzplan und muss in voller Entsprechung zu diesen Teilplänen stehen. Die Erstellung der Planbilanz erfolgt simultan mit der Erstellung des Finanzplanes, da sich jede Veränderung der einzelnen Vermögens- und Schuldposten auf deren Endbestand auswirkt.

Da die Budgeterstellung in der Regel noch im alten Jahr erfolgt, sind zu diesem Zeitpunkt die Aktiva und Passiva zu Beginn der Planperiode noch nicht gegeben. Man wird in diesen Fällen zunächst die voraussichtlichen Werte der Eröffnungsbilanz der Planperiode einsetzen und diese als Ausgangswerte für die Erstellung der Planschlussbilanz heranziehen. Nach Erstellung der tatsächlichen Eröffnungsbilanz des Planjahres wird eine Korrektur der vorläufigen Eröffnungsbilanz durchgeführt, die auch zu einer Änderung der Planschlussbilanz, nicht aber des Finanzplanes führt. Es besteht auch die Möglichkeit, mit der Erstellung der Planbilanz bis nach Fertigstellung der Eröffnungsbilanz der Planperiode zu warten.

---

<sup>53</sup> EGGER, A.; WINTERHELLER, M. (2004), S. 66

## 5. Testen von Software

In den letzten Jahren hat Software eine enorme Verbreitung gefunden. Es gibt kaum noch Geräte bzw. Anlagen in denen die Steuerung nicht mittels einer Software realisiert wird. Software übernimmt somit einen ganz entscheidenden Anteil am Funktionieren der Geräte und Anlagen. In allen Bereichen ist die Qualität der Software zum entscheidenden Faktor für den Erfolg von Produkten oder Unternehmen geworden. Wird Software nicht nur als Mittel zum Zweck für das Funktionieren von Maschinen eingesetzt, sondern nimmt Software die Funktion einer Ware ein, welche hergestellt und abgesetzt wird, so muss eine verbesserte Qualität der Softwaresysteme und des Softwareentwicklungsprozesses angestrebt werden.

Das Testen von Software zielt darauf ab, die Qualität von Softwaresystemen zu erhöhen. Aus diesem Grund wird dem Prüfen und Testen in der Praxis ein hoher Stellenwert zugeschrieben.<sup>54</sup>

### 5.1. Grundlagen zum Testen von Software

In Industriebetrieben werden die entstehenden Teilprodukte und daraus resultierende Endprodukte daraufhin kontrolliert, ob sie den gestellten Anforderungen entsprechen. Diese Vorgangsweise bei der Herstellung eines Industrieproduktes trifft entsprechend für die Produktion bzw. Entwicklung von Software zu. Das Testen von Software dient durch die Identifizierung von Defekten und deren anschließender Beseitigung durch das Debugging zur Steigerung der Softwarequalität. Die Testfälle sollen so gewählt werden, dass sie weitgehend der späteren Benutzung der Software entsprechen.<sup>55</sup>

Werden Kosten von Software-Entwicklungen analysiert kommt man zu dem Schluss, dass der weitaus größte Anteil der Aufwendungen während der Wartungsphase eines im Markt

---

<sup>54</sup> Vgl. FRÜHAUF, K.; LUDEWIG, J.; SANDMAYR, H. (2006), S. 20

<sup>55</sup> Vgl. SPILLNER, A.; LINZ, T. (2010), S. 6

befindlichen Produkts entsteht. Dies ist meist eine Folge der unzureichenden Qualität der Software. Fehler, welche im Zuge der Software-Erstellung entstehen, sind Ursachen für die mangelnde Qualität. Es existiert jedoch kein Verfahren, welches garantieren kann, dass fehlerfreie Produkte entstehen. Aus diesem Grund ist die Qualität mit analytischen Mitteln zu prüfen. Eine zentrale Stellung in der Entwicklung nimmt daher die Software-Prüfung ein. Das erstellte Programm wird gestartet und einige Eingaben werden vorgenommen, um sich die erzeugten Ausgaben anzusehen und sie mit den Erwartungen zu vergleichen.<sup>56</sup>

### 5.1.1. Definition des Fehlerbegriffes

In fast jedem Buch über Software-Tests findet man abweichende Definitionen vom Begriff des Fehlers. Eine Kernaussage haben jedoch alle Beschreibungen gemeinsam: Ein Fehler ist die Abweichung eines erwarteten Verhaltens zu dem tatsächlichen Verhalten eines Programms. Die Gründe für das Fehlverhalten können verschiedener Natur sein.

Der Trainer und Berater auf dem Gebiet der Softwarequalität und des Softwaretestens UWE VIGENSCHOW hat in seinem Buch „Objektorientiertes Testen und Testautomatisierung in der Praxis“ folgende Unterscheidungen getroffen:<sup>57</sup>

- **Fehlhandlung**

Darunter ist die menschliche Handlung des Entwicklers zu verstehen, die zu einem Fehlerzustand in der Software führt. Auch die menschliche Handlung eines Anwenders, die ein unerwünschtes Ergebnis zur Folge hat (= Fehlbedienung), wird unter dem Begriff Fehlhandlung eingeordnet.

- **Fehlerzustand**

Ist die Ursache für eine Fehlerwirkung ein inkorrektes Teilprogramm, eine inkorrekte Anweisung oder Datendefinition, so wird von einem Fehlerzustand gesprochen. Der Zustand eines Softwareproduktes oder einer seiner

---

<sup>56</sup> Vgl. LIGGESMEYER, P. (2009), S. 3f.

<sup>57</sup> Vgl. VIGENSCHOW, U. (2005), S. 18f.

Komponenten, der unter spezifischen Bedingungen, z.B. unter hoher Belastung, eine geforderte Funktion des Produkts beeinträchtigen kann bzw. zu einer Fehlerwirkung führt, wird ebenfalls als Fehlerzustand bezeichnet.

- **Fehlerwirkung**

Unter Fehlerwirkung wird die Wirkung eines Fehlerzustands, die bei der Ausführung des Testobjekts nach außen in Erscheinung tritt verstanden. Auch Abweichungen zwischen dem spezifizierten Sollwert und dem beobachteten Istwert liegen Fehlerwirkungen zu Grunde.

Die im Zuge der Diplomarbeit erstellte Testsoftware analysiert Fehlerwirkungen des untersuchten Programms Professional Planner.

### **5.1.2. Was versteht man unter Testen von Software**

Testen ist der Prozess, ein Programm mit der Absicht auszuführen, Fehler zu finden.<sup>58</sup>

Das Testen von Software ist ein aktiver Prozess. Treten beim Ausführen von Tests keine Fehler auf, kann man nicht davon ausgehen, dass keine Fehler vorhanden sind – es ist einfach nicht gelungen die vermuteten Fehler in der Software zu finden.<sup>59</sup>

### **5.1.3. Warum soll man testen?**

Seit 40 Jahren werden Software-Entwicklungstechnologien versprochen, welche eine Fehlerfreiheit gewährleisten oder zumindest die Fehleranzahl drastisch reduzieren. Die Fehlerrate pro 1.000 Anweisungen ist jedoch seit der Assembler-Programmierung nicht besser geworden. Im Jahr 2000 wurde für die Automobil- und Flugzeug-Branche in den USA ein Schaden durch fehlerhafte Software von 1,8 Milliarden US-Dollar errechnet. Das Hauptproblem in der Software-Entwicklung ist der Faktor Mensch. Der Mensch ist in seiner

---

<sup>58</sup> MYERS, G.J. (1989), S. 4

<sup>59</sup> Vgl. PARRINGTON, N.; ROPER, M. (1991), S. 1

genetischen Verfassung für den Entwurf und die Programmierung komplexer Softwaresysteme nicht geeignet. Die Erklärung dafür, dass die Fehlerquote bei Softwaresystemen der Luft- und Raumfahrt sowie der Rüstungsindustrie so gering ist, liegt darin, dass diese Systeme mehrfach gesiebt, geprüft und getestet wurden.<sup>60</sup>

#### 5.1.4. Allgemeine Prinzipien des Softwaretestens

Folgende Grundsätze zum Testen haben sich in den letzten 40 Jahren herauskristallisiert:<sup>61</sup>

- **Testen zeigt die Anwesenheit von Fehlern**

Durch das Testen wird das Vorhandensein von Fehlerwirkungen nachgewiesen. Mit Testen lässt sich nicht beweisen, dass keine Fehlerzustände im Testobjekt vorhanden sind. Auch wenn keine Fehlerwirkungen im Test aufgezeigt wurden, ist dies kein Nachweis für Fehlerfreiheit.

- **Vollständiges Testen ist nicht möglich**

Alle möglichen Eingabewerte und deren Kombinationen unter Berücksichtigung aller unterschiedlichen Vorbedingungen zu testen ist nicht durchführbar, Ausnahmen bilden hier sehr triviale Testobjekte. Tests sind immer nur Stichproben.

- **Mit dem Testen frühzeitig beginnen**

Durch frühzeitiges Prüfen werden Fehler frühzeitig erkannt. Daher sollen Testaktivitäten im System- oder Softwarelebenszyklus so früh wie möglich beginnen.

- **Häufung von Fehlern**

Es kommt sehr oft vor, dass sich in nur wenigen Teilen eines Testobjektes die meisten Fehlerwirkungen befinden. Von einer Gleichverteilung der Fehlerzustände

---

<sup>60</sup> Vgl. SNEED, H.; BAUMGARTNER, M.; SEIDL, R. (2007), S. 7ff.

<sup>61</sup> Vgl. SPILLNER, A.; LINZ, T. (2010), S. 36f.

im Testobjekt ist nicht auszugehen. Dort, wo Fehler nachgewiesen wurden, finden sich meist noch weitere.

- **Zunehmende Testresistenz**

Immer gleiche Tests decken keine neuen Fehlerwirkungen auf. Damit die Effektivität der Tests nicht absinkt, sind die vorhandenen Testfälle regelmäßig zu prüfen und durch neue oder modifizierte Testfälle zu ergänzen.

- **Testen ist abhängig vom Umfeld**

Keine zwei Systeme sind auf die exakt gleiche Art und Weise zu testen. Daher ist das Testen an das Einsatzgebiet und Umfeld des zu prüfenden Systems anzupassen.

### **5.1.5. Wechselwirkung von Testen und Qualität der Software**

Der Testprozess stellt einen integralen Bestandteil jedes Qualitätsprogramms dar. Um sicherzustellen, dass ein Produkt den Anforderungen entspricht, muss es vor dem Hintergrund dieser Anforderungen getestet werden. Die Anforderungen müssen klar definiert und testfähig sein um messen zu können, ob das Produkt den Anforderungen entspricht. Man sucht nicht nur aktiv nach Fehlern in Produkten, sondern ist auch bemüht Fehler gänzlich zu vermeiden. Wenn das Bemühen, Fehlern vorzubeugen, bei allen Arbeiten im Vordergrund steht, wenn Vermutungen von vornherein ausgeschlossen werden können, wenn in jeder Arbeitsstufe Tests ausgeführt werden, dann wird das Ergebnis ein Qualitätsprodukt sein.<sup>62</sup>

---

<sup>62</sup> Vgl. PARRINGTON, N.; ROPER, M. (1991), S. 12f.

## **5.2. Zum Testprozess im Allgemeinen**

Beim traditionellen Software-Testen stand das Testen des Programmcodes im Vordergrund. Heutzutage stellt das Testen des Programmcodes nur einen Teil des Testprozesses dar. Die klassische Analyse der Anforderungen ist in jedem Projekt der erste Schritt. Auch bei gegenwärtigen Testprojekten stellen die Anforderungen den Ausgangspunkt dar. Die Testanforderungen werden aus den allgemeinen Anforderungen an die Software herausgefiltert und abgeleitet. Gegenwärtig werden in jeder Phase des Lebenszyklus Tests ausgeführt. Ziel ist es, ein vollkommen ausgetestetes Produkt in die nächste Phase zu überführen.<sup>63</sup>

### **5.2.1. Teilprozess: Testplanung**

Zwei Dokumente bieten sich zur Erstellung des Testplans als Informationsquelle an: Die Spezifikation, welche alle Forderungen an die Software enthalten sollte und der Softwareentwicklungsplan. Neben den Anforderungen an die Software, welche aus der Spezifikation gewonnen werden, benötigt man weitere Informationen, die aus dem Softwareentwicklungsplan herauszulesen sind. Die Frage nach welcher Methode beim Design vorgegangen wird (Top-down oder Bottom-up) muss aus diesem Entwicklungsplan heraus beantwortet werden. Diese Frage ist für den Test deswegen interessant, weil man daraus ableiten kann, wann mit dem Testen der Software begonnen werden kann. Der Zeitplan gibt darüber hinaus Aufschluss ob für die Softwareentwicklung und die Testarbeiten realistische Annahmen getroffen wurden.<sup>64</sup>

### **5.2.2. Teilprozess: Testanalyse und Testdesign**

Testfälle lassen sich aus den Ergebnissen der Anforderungsanalyse ableiten. Die Testfälle sollen das gesamte System ausreichend testen, aber auch für Integrationstests und Tests

---

<sup>63</sup> Vgl. VIGENSCHOW, U. (2005), S. 17f.

<sup>64</sup> Vgl. THALLER, G.E. (2002), S. 35

einzelner Komponenten herangezogen werden. In manchen Fällen können die Testfälle sogar für einzelne Klassen herunter gebrochen werden. Die zu programmierenden Abläufe können in Ablaufdiagrammen erfasst werden und bieten somit eine hervorragende Basis zur Entwicklung fachlicher Testfälle. Als Analysedokument sollten die Anwendungsfälle bereits früh im Testprozess zur Verfügung stehen. Dadurch kann eine Analyse aus Testsicht früh im Projektverlauf erfolgen.<sup>65</sup>

### **5.2.3. Teilprozess: Testrealisierung und Testdurchführung**

Testinfrastruktur und Testrahmen sind im Detail zur Testdurchführung zu realisieren. Es ist erforderlich den konkreten Ablauf der Tests festzulegen. Einzelne Testfälle werden zu Testsequenzen bzw. zu Testszenarien gruppiert, um die Testfälle effektiv durchführen zu können und eine übersichtliche Struktur der Testfälle zu erhalten. Sind alle vorbereitenden Arbeiten abgeschlossen, kann nach der Übergabe der zu testenden Systemteile an die Tester, mit der Testdurchführung begonnen werden. Testläufe werden durchgeführt und protokolliert. Es ist genau festzuhalten, welche Testläufe mit welchem Ergebnis durchgeführt wurden.<sup>66</sup>

### **5.2.4. Teilprozess: Testauswertung und Berichte**

Im Zuge der Testauswertung wird überprüft, ob die in der Planung festgelegten Testergebnisse erfüllt worden sind.

Bei den Testergebnissen handelt es sich um jene Produkte, die das Testteam im Laufe des Tests zu liefern hat. Anhand der Ergebnisse wird der Fortschritt des Tests gemessen. Die

---

<sup>65</sup> Vgl. VIGENSCHOW, U. (2005), S. 48f.

<sup>66</sup> Vgl. SPILLNER, A.; LINZ, T. (2010), S. 27ff.

Liste beginnt mit dem Testplan und endet mit dem Testabschlussbericht. Folgende Ergebnisse sind das Minimum, das ein Testprojekt anstreben sollte:<sup>67</sup>

- Testplan
- Testentwurf bzw. Testkonzept
- Testfallspezifikation
- Testskripte
- Testausführungsprotokolle
- Fehlerberichte
- Testabschlussbericht

### **5.3. Grundsätzliche Ausprägungen von Tests**

Im Allgemeinen geht es beim Testen von Software darum, zu überprüfen, ob das Programm sich so verhält, wie spezifiziert bzw. ob die Funktionalität korrekt bereitgestellt wird.

Natürlich können auch noch weitere Fragen in Bezug auf die Korrektheit und Fehlerfreiheit des Programms gestellt werden, zum Beispiel nach dem Verhalten der Software bei hoher Last. Man kann auch untersuchen wie sich das Programm bei großen zu verarbeitenden Datenmengen sowie unter verschiedenen Hardwarekonfigurationen verhält.

Aus diesen Überlegungen heraus können die verschiedenartigen Tests in mehrere Gruppen unterteilt werden. Die für diese Arbeit in Frage kommenden Testarten werden in den Kapiteln 5.3.1. Funktionale Tests und 5.3.2. Nicht funktionale Tests aufbereitet.

#### **5.3.1. Funktionale Tests**

Die funktionalen Tests subsumieren alle Testmethoden, mittels derer das von außen sichtbare Ein-/Ausgabeverhalten eines Testobjekts geprüft wird. Funktionale Anforderungen spezifizieren das Verhalten, welches das System oder Systemteile erbringen müssen. Die Umsetzung bildet die Voraussetzung dafür, dass das System einsetzbar ist.<sup>68</sup>

---

<sup>67</sup> Vgl. SNEED, H.; BAUMGARTNER, M.; SEIDL, R. (2007), S. 67

<sup>68</sup> Vgl. SPILLNER, A.; LINZ, T. (2010), S. 72

„Wichtig ist vor allem, dass man die Funktionen der Software dort testet, wo sie zum ersten Mal sinnvoll getestet werden können.“<sup>69</sup>

Einen Funktionstest durchzuführen bedeutet, ein Programm anhand der Spezifikationen zu testen. Außer Acht gelassen wird die interne Funktionsweise des Programms. Deshalb wird der Funktionstest auch als Black-Box-Test bezeichnet. Wie Abbildung 6 zeigt wird das Programm als Black-Box betrachtet, deren Funktionalität durch Beobachten der Ausgaben zu den entsprechenden Eingaben überprüft wird.



Abbildung 6: Black-Box-Test<sup>70</sup>

Der Funktionstest gibt darüber Aufschluss, ob das richtige Produkt erstellt wurde. Die Tests werden auf Grundlage der Spezifikationen konzipiert. Das Programm wird mit Eingabedaten ausgeführt und die Ergebnisse beobachtet.<sup>71</sup>

### 5.3.2. Nicht funktionale Tests

Die nicht funktionalen Anforderungen beschreiben Attribute des funktionalen Verhaltens, also „wie gut“ oder mit welcher Qualität das System funktionieren soll. Die Umsetzung dieser Anforderungen beeinflusst sehr stark, wie die Kunden bzw. Anwender mit dem Produkt zufrieden sind. Merkmale sind etwa: Zuverlässigkeit, Benutzbarkeit und Effizienz. Auch die Änderbarkeit und Anpassungsfähigkeit nehmen Einfluss auf die

---

<sup>69</sup> THALLER, G.E. (2002), S. 122

<sup>70</sup> Quelle: Eigene Darstellung

<sup>71</sup> Vgl. PARRINGTON, N.; ROPER, M. (1991), S. 51

Kundenzufriedenheit. Umso einfacher und schneller ein System an geänderte Anforderungen anpassbar ist, desto zufriedener wird der Kunde und Anwender sein.<sup>72</sup>

Die im Zuge dieser Diplomarbeit entworfene Testsoftware bedient sich neben dem Funktionstest auch zwei nicht funktionalen Testarten, dem Volume Test und dem Stress Test.

#### **5.3.2.1. Volume Test**

Der Volume Test ist darauf angelegt, die Grenzen der Belastbarkeit eines Programms auszuloten. Es geht konkret darum, das System durch große Datenmengen zu belasten. In vielen Fällen ist die Software so ausgelegt, dass vorhandene Grenzen der Belastbarkeit in der Praxis nie erreicht, geschweige denn überschritten werden.<sup>73</sup>

#### **5.3.2.2. Stress Test**

Der Stress Test hat eine gewisse Ähnlichkeit mit dem Volume Test, er unterscheidet sich jedoch darin, dass das Programm innerhalb eines kurzen Zeitraums einer hohen Belastung ausgesetzt wird. Als Beispiel könnte etwa ein Test dienen, bei dem eine große Anzahl an Benutzern versucht sich in das System einzuloggen.<sup>74</sup>

### **5.4. Testautomatisation**

Vollautomatisierte Tests laufen ohne Interaktion mit dem Endbenutzer im Hintergrund ab. Diese Art der Tests wird durch ihre Eingabedaten gesteuert. Diese Daten werden aus Bewegungsdaten oder aus Datenbanktabellen gelesen. Die Aufgabe der Tester ist es, diese Dateien oder Datenbanken bereitzustellen. Auf der Ausgabeseite werden andere Dateien

---

<sup>72</sup> Vgl. SPILLNER, A.; LINZ, T. (2010), S. 74

<sup>73</sup> Vgl. THALLER, G.E. (2002), S. 124

<sup>74</sup> Vgl. THALLER, G.E. (2002), S. 128

oder Datenbanken erstellt oder geändert. Sehr oft werden auch Berichte oder Listen generiert. Die Tester kontrollieren am Ende des automatisierten Tests die Ergebnisobjekte.<sup>75</sup>

#### 5.4.1. Motivation für Testautomatisation

Ziel einer Testautomatisierung ist es, Tests effizient und langfristig ökonomisch durchführen zu können. Die Qualität des Testfalls wird durch seine Automatisierung nicht erhöht.<sup>76</sup>

Nicht jeder Test kann mit Erfolg automatisiert werden. Die Automatisierung des Testbetriebes ist nicht ohne beträchtliche Investitionen in Testverfahren, Werkzeuge und Schulung der Mitarbeiter möglich. Organisationen, die sich intensiv mit der Fragestellung nach Automatisierung auseinandersetzen, können durch Test-Automatisierung erhebliche Einsparungen, wie nachstehende Tabelle an einem Beispiel zeigt, erzielen.<sup>77</sup>

<b>Tätigkeit</b>	<b>Manueller Test</b>	<b>Automatischer Test</b>
Kosten für das Design der Testfälle in US \$	6.000,00	6.000,00
Kosten für den Kauf eines Tools in US \$	-	5.000,00
Kosten für die Automatisierung der Testfälle	-	11.000,00
Gesamtkosten der Automatisierung	-	16.000,00
Kosten für die Ausführung einer Reihe von Testfällen (Suite) in US \$	5.000,00	1.000,00
Zahl der Ausführung der Test-Suite pro Release der Software	3	3
Einsparungen pro Release der Software in US \$	21.000,00	9.000,00
Einsparungen pro Release in US \$		12.000,00
Anzahl der Releases pro Kalenderjahr	4	4
Einsparungen pro Jahr in US \$		32.000,00
Return on Investment (ROI)		200 %

**Tabelle 3: Einsparungen durch Test-Automatisierung**<sup>78</sup>

<sup>75</sup> Vgl. SNEED, H.; BAUMGARTNER, M.; SEIDL, R. (2007), S. 137

<sup>76</sup> Vgl. VIGENSCHOW, U. (2005), S. 159f.

<sup>77</sup> Vgl. THALLER, G.E. (2002), S. 228

<sup>78</sup> Vgl. THALLER, G.E. (2002), S. 239

### 5.4.2. Einsatz von Werkzeugen

Ein Testmonitor ist erforderlich, um den automatisierten Testdurchlauf zu überwachen. In vielen Fällen zeichnet ein derartiger Testmonitor die Verwendung der Funktionen in den Modulen in der abgearbeiteten Reihenfolge auf. Diese Ablaufdaten bzw. Ablaufpfade werden in Protokollen gespeichert. Ein Testvalidator prüft die Ausgabedateien, Datenbanken und Berichte gegen die erwarteten Ergebnisse. Ausgabewerte, die mit den Sollwerten nicht übereinstimmen, werden in einem Abweichungsbericht ausgewiesen. Der Tester ist für die Weiterverarbeitung dieser Abweichungsberichte verantwortlich. Er prüft warum die falschen Ergebnisse zu Stande gekommen sind und zeigt mögliche Ursachen für die Abweichungen auf.<sup>79</sup>

### 5.4.3. Einsatzgebiete automatisierter Softwaretests<sup>80</sup>

Mit Hilfe bestimmter Tools, wie beispielsweise Mercury Interactive WinRunner, können Tests automatisiert werden. Diese Tests sind zumeist Black-Box-Tests, also spezifikationsorientiert. Ein klassisches Einsatzgebiet ergibt sich in der Reproduktion aufwendig zu erstellender Fehler. Das zu testende Programm kann zum Beispiel ohne Probleme durch alle Tests kommen, in der Praxis jedoch nach mehrstündiger Arbeit häufig abstürzen. Um auf den Fehler zu kommen muss das typische Benutzerverhalten nachgebildet werden. Dafür wird der Benutzerablauf automatisiert und diese Testroutinen automatisch ausgeführt. Spezielle Mechanismen, wie zum Beispiel Debugging-Tools, protokollieren Eingabedaten und Ausgabedaten mit, um herauszufinden, welche Kombination zu einem Absturz oder Fehler führt.

Ein weiterer Bereich, für den sich die Testautomatisation sehr gut eignet, ist der des Stresstests. Diese Art der Tests wird nicht tatsächlich von Testern durchgeführt sondern nur simuliert. Der Wert dieser Tests kann kaum überwertet werden. Einerseits erhält man so realistische Angaben zum Zeitverhalten der Anwendung, andererseits kann sich das

---

<sup>79</sup> Vgl. SNEED, H.; BAUMGARTNER, M.; SEIDL, R. (2007), S. 138

<sup>80</sup> Vgl. VIGENSCHOW, U. (2005), S. 191ff.

Verhalten einer Anwendung stark unterscheiden, wenn nur wenige oder aber sehr viele Anwender darauf zugreifen.

Das Verhalten der Software kann auch in einem Dauertest analysiert werden. Dazu werden die Tests in einer Schleife abgearbeitet. Dieser Testlauf kann je nach Anforderung über Nacht oder über mehrere Tage oder Wochen laufen.

## 6. Der Software-Entwicklungsprozess

Softwareentwicklungsmodelle und Entwicklungsprozesse werden eingesetzt, um eine strukturierte und steuerbare Softwareentwicklung durchzuführen. Es existiert eine Reihe von unterschiedlichen Modellen. Jedes Modell definiert eine Systematik zur geordneten Projektabwicklung.<sup>81</sup>

Software-Entwicklungsprozesse dienen der Steuerung einer Softwareentwicklung. Angefangen bei der Konzeption bis zum Einsatz im Echtbetrieb inklusive der im Echtbetrieb anfallenden Änderungen einer Software sind spezielle Vorgehensmodelle (Prozessmodelle) definiert.

### 6.1. Prozessmodelle in der Softwareentwicklung

Durch eine steigende Komplexität der zu entwickelnden Software-Systeme hat das Thema Vorgehensmodelle in der Software-Entwicklung an Bedeutung gewonnen. In den letzten Jahren haben Publikationen über spezialisierte Vorgehensmodelle zu einem unübersichtlichen Wachstum geführt.<sup>82</sup>

Wie bereits erwähnt, existiert mittlerweile eine Vielzahl an Software-Entwicklungsmodellen. Im folgenden Abschnitt werden zwei grundlegende Vorgehensmodelle erklärt.

Das Wasserfallmodell ist der Klassiker unter den Vorgehensmodellen. Das Modell wurde bereits in den 1970er Jahren von J. Royce veröffentlicht. Namensgebend ist, dass die

---

<sup>81</sup> Vgl. SPILLNER, A.; LINZ, T. (2010), S. 18

<sup>82</sup> Vgl. BUNSE, C.; KNETHEN, A. (2008), S. 3f.

sequenziellen Schritte des Lebenszyklus eines Software-Produkts in Form eines Wasserfalls, wie in Abbildung 7, dargestellt werden.

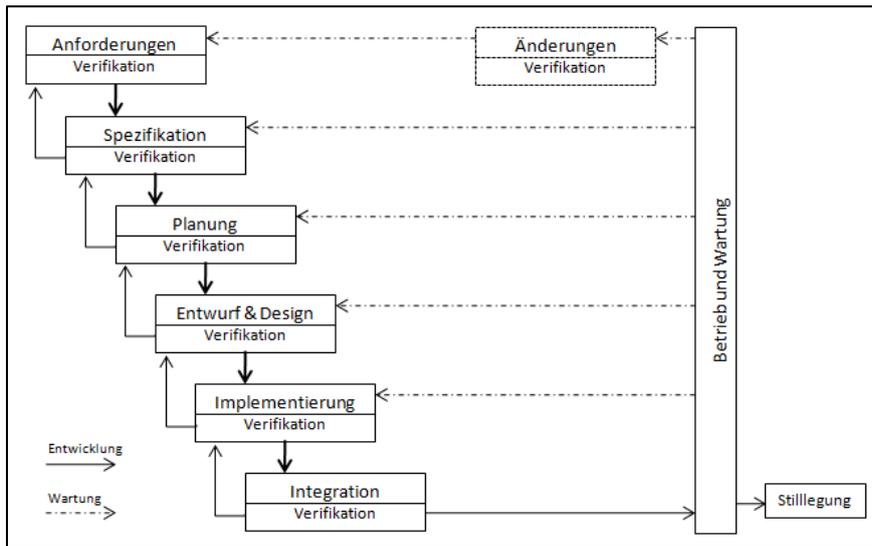


Abbildung 7: Wasserfallmodell<sup>83</sup>

Das V-Modell (Abbildung 8) basiert ebenfalls auf dem Software-Lebenszyklus und umfasst die wesentlichen Schritte von der Konzeptphase eines Software-Projekts, über Planung, Entwurf, Design und Implementierung bis zu Test, Wartung und Weiterentwicklung. Das V-Modell deckt, so wie das Wasserfallmodell, einen Großteil des Software-Lebenszyklusses ab.<sup>84</sup>

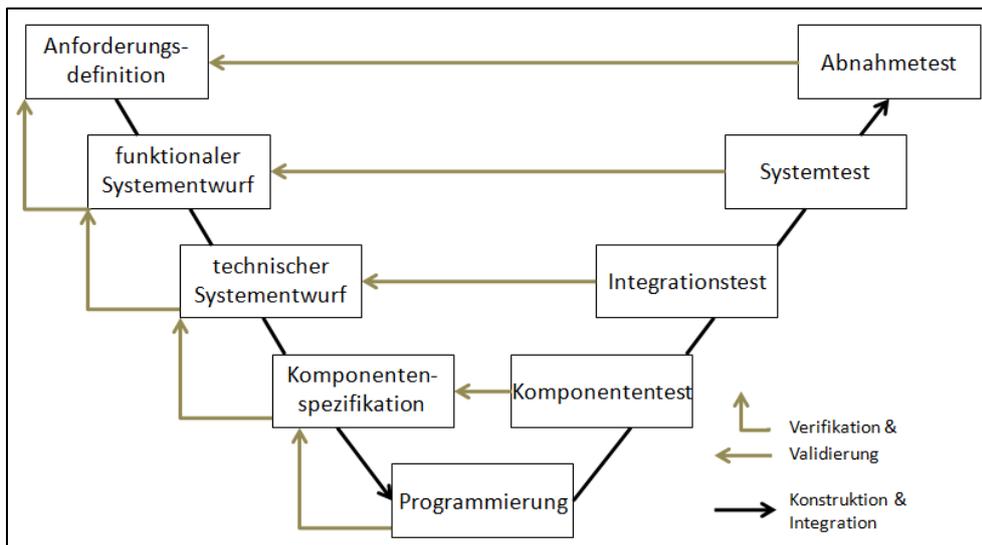


Abbildung 8: V-Modell<sup>85</sup>

<sup>83</sup> Vgl. SCHATTEN, A.; BIFFL, S.; DEMOLSKY, M.; GOSTISCHA-FRANTA, E.; ÖSTREICHER, T.; WINKLER, D. (2010), S. 49

<sup>84</sup> Vgl. SCHATTEN, A.; BIFFL, S.; DEMOLSKY, M.; GOSTISCHA-FRANTA, E.; ÖSTREICHER, T.; WINKLER, D. (2010), S. 49

<sup>85</sup> Vgl. SPILLNER, A.; LINZ, T. (2010), S. 42

Die beiden angesprochenen Prozessmodelle sind für große Softwareprojekte geeignet und haben sich in den letzten Jahrzehnten in der Softwareentwicklung bewährt. Wie bereits erwähnt hat sich, oft abgeleitet von diesen Vorgehensmodellen, eine große Vielfalt an neuen, modernen Herangehensweisen entwickelt.

Für kleine Projekte, mit wechselnden Anforderungen und starker Kundenbindung, wird als alternatives Prozessmodell das so genannte Extreme Programming diskutiert. Dieses Prozessmodell sieht eine Entwicklung in schnellen Iterationen vor. Durch permanentes Validieren mit den Kunden wird ein erwartungskonformes Software-Produkt garantiert. Der sinnvolle Einsatz von Extreme Programming ist an die Erfüllung bestimmter Voraussetzungen gebunden. Das Projekt muss hinreichend klein sein und der Kunde muss bereit sein aktiv mitzuarbeiten.<sup>86</sup>

Extreme Programming (XP) gehört zu den leichtgewichtigen Vorgehensmodellen der Softwareentwicklung. XP wurde Ende der 1990er Jahre von Kent Beck entwickelt und gehört inzwischen zu den populärsten objektorientierten Vorgehensmodellen. Das Grundprinzip ist die rasche Erstellung eines ausführbaren Systems.<sup>87</sup>

## **6.2. Das ausgewählte Vorgehensmodell Extreme Programming**

XP setzt auf Kommunikation, Teamarbeit und Offenheit. Mit Kommunikation ist die ständige Interaktion aller Projektbeteiligten gemeint. Das Zusammenspiel aller am Projekt beteiligten Entwickler steht im Vordergrund. Da die Programmierarbeit für die vorliegende Arbeit von einer Person bewältigt wurde, ist hier im konkreten Fall nicht die Kommunikation zwischen den Programmierern gemeint, sondern das ständige Besprechen des Fortschritts mit Mitarbeitern des Kooperationspartners, der Firma WINTERHELLER software GmbH. Auch der Teamgedanke des XP wurde im Projekt nicht 1:1 umgesetzt, da das Entwicklerteam nur aus einer Person bestand. Die frühzeitige Einbeziehung des Benutzers bzw. Kunden war jedoch ein ausschlaggebender Punkt während der Implementierungsphase. Auch die typischen

---

<sup>86</sup> Vgl. LISSGESMEYER, P. (2009), S. 364

<sup>87</sup> Vgl. BUNSE, C.; KNETHEN, A. (2008), S. 115

Phasen des XP, wie die Planungsphase und die iterative Phase zur Realisierung des Systems wurde nach dem XP-Vorbild umgesetzt. Für die Wahl von XP sprachen die kurze Entwicklungszeit und die laufend verändernden Anforderungen an die Software.

Das Konzept von Extreme Programming konnte aufgrund der Gegebenheiten nicht vollständig umgesetzt werden. Adaptionen hinsichtlich der Größe des Projektteams und der Vorgehensweisen waren nötig. So bestand das Entwicklerteam nicht aus idealerweise 12 bis 14 Personen, sondern aus einem Softwareentwickler. Die User Stories und Testszenarien wurden laut XP geplant und umgesetzt. Das Erstellen einer prototypischen Architektur und eines Release-Plans wurde aus XP Sicht ebenfalls verwirklicht.

Das Prototyping ist eine Methode zur Systementwicklung, bei der möglichst früh in einem Pilotprojekt eine erste vereinfachte Version des Programms (Prototyp) realisiert wird, um in wichtigen, aber auch offenen Bereichen (z.B. Anwenderschnittstellen, Datenstrukturen, Design) Erfahrungen zu sammeln.<sup>88</sup>

---

<sup>88</sup> Vgl. VETTER, M. (1988), S. 146

## **7. Ist-Analyse – Ausgangssituation**

Die Ist-Analyse stellt das Herzstück einer jeden Projektarbeit dar. Sie ist die Grundlage um Verbesserungspotenziale zu entwickeln und einen etwaigen Handlungsbedarf offen zu legen. In der vorliegenden Projektarbeit dient die Ist-Analyse der Erhebung der aktuellen Situation in der Testabteilung des Kooperationspartners WINTERHELLER software GmbH.

Für die Erstellung der Testsoftware werden die bestehenden Abläufe in der Testabteilung analysiert, um davon ein Anforderungsprofil an das Programm ableiten zu können. In der vorliegenden Arbeit dient die Ist-Analyse zur Beantwortung folgender Fragen:

1. Wie ist die Testabteilung organisiert?
2. Aus welchen Bestandteilen besteht der Testablauf?
3. Welche Charakteristika weisen die Testdaten auf?

### **7.1. Organisation der Testabteilung**

Die Testabteilung der Firma Winterheller setzt sich aus vier Mitarbeitern zusammen und ist organisatorisch der Entwicklungsabteilung unterstellt. Zwei der vier Mitarbeiter sind vollzeitbeschäftigt, die beiden anderen auf Halbtagsbasis für das Testen der Softwarepakete angestellt.

Die Abteilung hängt nicht nur organisatorisch, sondern auch räumlich mit der Entwicklungsabteilung zusammen. Da keine räumliche Trennung besteht, können Informationen sehr schnell und unbürokratisch zwischen den beiden Abteilungen ausgetauscht werden. Die Gefahr besteht jedoch, dass die Weitergabe der Informationen nicht immer auf offiziellen und damit nachvollziehbaren Weg geschieht. Die Ablauforganisation sieht eine Dokumentation der gefundenen Fehler in elektronischer Form vor. Das genaue Dokumentieren einer Fehlersituation und dessen Beschreibung macht die Nachvollziehbarkeit von Fehlern für die Entwicklungsabteilung möglich. Auch der

Arbeitsfortschritt kann durch elektronische Auswertungen sehr schnell erhoben und nachgeprüft werden.

Die Softwarepakete werden nach dem Fortschritt der Entwicklungsabteilung getestet. Neue Versionen bzw. Releases benötigen im Gegensatz zu kleineren Anpassungen mehrere Tage um vollständig getestet werden zu können.

Die Testabteilung wird in Entwicklungsmeetings mit einbezogen, um zu jeder Zeit auf dem neuesten Stand zu bleiben. Dies ist sehr wichtig, da die Testabläufe koordiniert und geplant werden müssen.

Die folgenden Punkte beziehen sich auf das in dieser Arbeit behandelte Softwarepaket Professional Planner.

## 7.2. Bestandteile des Testablaufs

Der Testablauf unterliegt einem vordefinierten Schema. Wird eine neue Version getestet, geht die Testabteilung wie folgt vor:

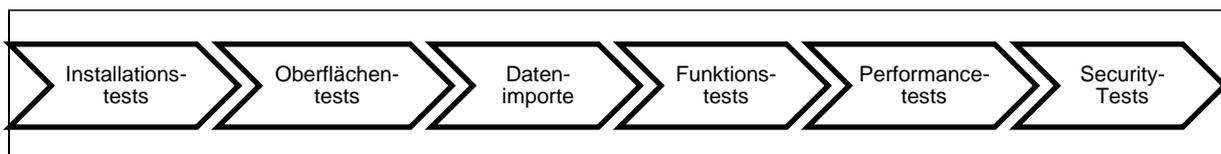


Abbildung 9: Testablauf<sup>89</sup>

1. Zu Beginn werden Installationstests durchgeführt. Dabei wird die Installationsroutine auf den unterstützten Systemen ausgeführt. Nach erfolgreicher Installation werden die Startmenüeinträge auf Korrektheit und das Vorhandensein der Dokumentation überprüft. Ein Starten der Software beschließt diesen ersten Testvorgang.
2. Anschließend an die Installationstests werden sogenannte Oberflächentests durchgeführt. Diese zweite Testphase zielt auf die korrekte und fehlerfreie Darstellung der Programmoberfläche ab. Konkret bedeutet dies zu überprüfen, ob

---

<sup>89</sup> Quelle: Eigene Darstellung

- die Menüs, Baumstrukturen, Befehle, Icons und Fensterdarstellungen auf den verschiedenen Betriebssystemen konsistent und ordnungsgemäß dargestellt werden.
3. Um die Funktionalität der Programme überprüfen zu können, müssen vorhandene Datensätze in die jeweilige Software importiert werden. Dieser Import der Daten stellt ebenfalls einen Teil der Testroutine dar. Nach erfolgreichem Einspielen der Daten werden diese auf Ihre Korrektheit überprüft.
  4. Wurden die Daten erfolgreich eingespielt, beginnen die Funktionstests. Im Zuge dieser Tests wird die Korrektheit der Business Content Library, kurz BCL genannt, getestet. Die BCL legt die grundlegenden rechnerischen und unternehmerischen Zusammenhänge innerhalb eines Unternehmensmodells fest. Die frühere Bezeichnung dafür war „Rechenschema“. Die Rechenlogik wird in einer externen Datei, der BCL, festgelegt und nicht starr in der Applikation hinterlegt. In der BCL werden die betriebswirtschaftlichen Zusammenhänge mit Hilfe sogenannter Strukturelemente abgebildet. Als betriebswirtschaftliche Basis für die BCL des Professional Planners dient dabei das Buch von EGGGER/WINTERHELLER: „Die kurzfristige Unternehmensplanung“. Um die BCL auf Korrektheit zu überprüfen, werden Standardvorgänge von den einzelnen Testern durchgespielt und die Ergebnisse mit Referenzdaten verglichen. Handelt es sich dabei nicht um aufwendige Rechenlogiken, werden Individualtests durchgeführt und deren Ergebnisse von den Mitarbeitern der Testabteilung verifiziert. Komplexere Abläufe bzw. kompliziertere Rechenoperationen (zB Newton'sche Näherungsverfahren) werden anhand von Referenzdaten überprüft.
  5. Die letzte Phase umfasst die Performance- und Security-Tests. In dieser Testphase werden Leistungsfähigkeit, Effizienz, Durchsatz und sicherheitsrelevante Aspekte überprüft.<sup>90</sup>

Einzelne Funktionstests werden nach Bedarf durchgeführt und unterliegen keinem genauen Ablaufplan. Die Dokumentation der zu testenden Funktionalität und die Ergebnisse der Tests werden auch hier in elektronischer Form erfasst.

---

<sup>90</sup> M. Hirczi (persönliches Interview, 13.07.2010)

### **7.3. Charakteristika der Testdaten**

Die Testdaten beschreiben den Testfall. Diese Daten werden während des Testens eingegeben und die dadurch generierten Ausgabedaten werden mit den Solldaten verglichen. Die Solldaten werden entweder manuell erstellt oder mittels geeigneten Mechanismen automatisch generiert.

Für das Testen des Professional Planner werden Daten aus einem Standardfallbeispiel verwendet. Dieses Fallbeispiel wurde von Mitarbeitern der Firma Winterheller erarbeitet und dient ebenfalls als Schulungsbeispiel für Neukunden bzw. neue Mitarbeiter. Die erwarteten Ergebnisse wurden von den Mitarbeitern per Hand verifiziert. Die Tester arbeiten dieses Fallbeispiel durch und vergleichen Ergebnisse wie Deckungsbeiträge, Gewinn/Verlust und Cashflow mit den manuell errechneten Resultaten. Auch Einstellungen am Programm müssen für dieses Fallbeispiel vorgenommen und getestet werden.

Das Überprüfen der Referenzdaten mit den tatsächlichen Werten geschieht manuell. Das heißt, die Tester prüfen jeden einzelnen Wert im Programm. Ebenso werden die Einstellungen auf deren Richtigkeit und Vollständigkeit überprüft.

## **8. Entwicklung der Winterheller TestSuite**

Das vorangegangene Kapitel 7 bildet die Basis für den Entwurf der WINTERHELLER Testsoftware. Die Bestandteile des Testablaufes und die Charakteristika der Testdaten geben darüber Aufschluss, welches Verbesserungspotential durch eine geeignete Software ausgeschöpft werden kann. Da alle Tests einzeln von den Mitarbeitern der Testabteilung ausgeführt werden, liegt hier ein enormes Potential für den Einsatz automatisierter Testvorgänge. Ziel wird es sein, der Testabteilung ein Testtool zur Verfügung zu stellen, mit dem einzelne Testvorgänge automatisiert ausgeführt werden können.

Wie schon zu Beginn von Kapitel 6 beschrieben, strukturieren Software-Entwicklungsprozesse das Software-Projekt und geben einen Überblick über den Projektfortschritt. So wie für alle Vorgehensmodelle galt auch für das vorliegende Projekt die Einteilung in die Anforderungserhebung, Entwurf- und Designphase, Implementierung, Validierung und in die Betriebnahme sowie die Wartung der fertiggestellten Software. Letzter Punkt, die Wartung, wurde für dieses Projekt, wegen des begrenzten Zeitrahmens der Diplomarbeit, nicht in das Vorgehensmodell mit einbezogen.

### **8.1. Festlegung von Zielen**

Die Festlegung von zeitlichen sowie inhaltlichen Zielen bildete den Start in das Projekt und ist mit der Anforderungserhebung gleichzusetzen. Dafür wurden als erstes der Einsatzzweck und die Funktionsweise des zu testenden Programms beleuchtet. Für den Entwurf eines automatisierten Testtools war es unabdingbar zu verstehen, wie der Untersuchungsgegenstand, im Konkreten der Professional Planner, funktioniert bzw. welche Aufgaben mittels der Software erledigt werden können. Nach einer Einarbeitungsphase in das Programm wurde in einem ersten Meeting mit der Entwicklungs- sowie Testabteilung mögliche Ziele für das Projekt erarbeitet und Grenzen definiert. Nach dieser ersten Anforderungserhebung wurden die zur Verfügung stehenden Schnittstellen des Professional

Planners analysiert. Diese geben Aufschluss darüber, welche Funktionalität durch das Testtool abgeprüft werden kann.

Nach einem zweiten Meeting mit Teilen der Entwicklungs- und Testabteilung wurden grobe Ziele definiert:

- Es sollte möglich sein Benutzereingaben anhand von Eingabedaten zu simulieren. Diese Eingabedaten werden in CSV- oder Excel-Dateien abgebildet.
- Von enormer Bedeutung ist es, die abgearbeiteten Aufgaben und Ergebnisse in einem Dokument, dem so genannten Testlog, festzuhalten. Darin sollen alle Tests und deren Ausgang aufgezeichnet werden.
- Die Tests sollten total selbstständig, zum Beispiel in der Nacht ohne weitere Benutzereingabe, ablaufen können.
- Das Testtool müsste im Stande sein, in weiteren Ausbaustufen die Programme Professional Treasury sowie Professional Consolidation bedienen zu können.

Nachdem die ersten Ziele aus der Anforderungserhebung erarbeitet wurden, konnte mit dem Entwurf und Design der Testsoftware begonnen werden.

## **8.2. Entwurf und Design der Testsoftware**

Unter Berücksichtigung der grob definierten Anforderungen wurde das erste Design der Software entworfen. Wie schon in Kapitel 6 erwähnt, ist das Extreme Programming von schnellen Entwicklungszeiten bzw. einer schnellen Prototypengenerierung geprägt. Bevor der erste Prototype erstellt werden konnte, wurden die Anforderungen strukturiert und in Anwendungsfälle überführt.

Mittels der TestSuite sollte es möglich sein, eine Sequenz von Benutzereingaben zu simulieren. Daraus entstand die Idee, Benutzereingaben zu gliedern und in abzuarbeitende Einheiten, sogenannten Tasks, einzuteilen. Eine Kategorisierung in technische und

betriebswirtschaftliche Tasks, wie Abbildung 10 zeigt, schien zu diesem Zeitpunkt als geeignetes Modell.

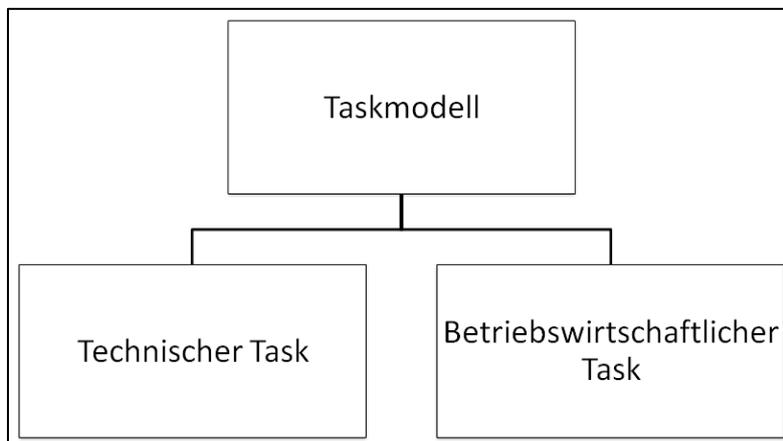


Abbildung 10: Taskmodell<sup>91</sup>

Unter dem Begriff Technischer Task sollten alle Vorgänge zusammengezogen werden, die mit technischen Aspekten, wie zum Beispiel die Installation des Programms, zu tun haben. Die betriebswirtschaftlichen Tasks umfassen alle diesbezüglichen Benutzereingaben. Somit war vorgegeben, dass Interaktionen des Benutzers mit der Software zu Tasks geformt und diese später im Testtool exekutiert werden. Dieses Modell wurde im Laufe der Arbeit verfeinert bzw. angepasst.

Einzelne Tasks sollten zu sogenannten Szenarien kombiniert werden, um komplexe Testvorgänge zu kreieren. Diese Szenarien bzw. die darin enthaltenen Tasks werden mit Eingabedaten und Referenzdaten belegt. Diese zusammengestellten Konfigurationen können mit allen notwendigen Daten als Szenarien gespeichert und später wiederverwendet werden.

Nachdem geklärt war, wie die grobe Gliederung der abzuarbeitenden Arbeitsanweisungen aussieht, wurde mit dem Design der ersten Prototypen begonnen. Wie bereits in Kapitel 6 erwähnt, stellt das Prototyping eine Methode zur Systementwicklung bereit, durch die so früh wie möglich eine erste vereinfachte Version (Prototyp) der angestrebten Anwendung

---

<sup>91</sup> Quelle: Eigene Darstellung

realisiert wird. Dieser Prototyp dient als Kommunikationsgegenstand zwischen Auftraggeber und Entwickler.<sup>92</sup>

Die Wahl der Entwicklungsumgebung fiel auf das Microsoft Visual Studio 2008. Als Programmiersprache einigte man sich auf C sharp (C#) in Verbindung mit der .Net-Plattform.

C# ist eine einfache, objektorientierte und typsichere Programmiersprache, die ihre Wurzeln in der C-Sprachfamilie hat.<sup>93</sup>

Diese Programmiersprache ist eine von Microsoft entwickelte Programmiersprache für die .Net-Plattform. Sie besitzt alle Eigenschaften um Programme schnell und effizient nach dem neuesten Stand der Softwaretechnik zu entwickeln.<sup>94</sup>

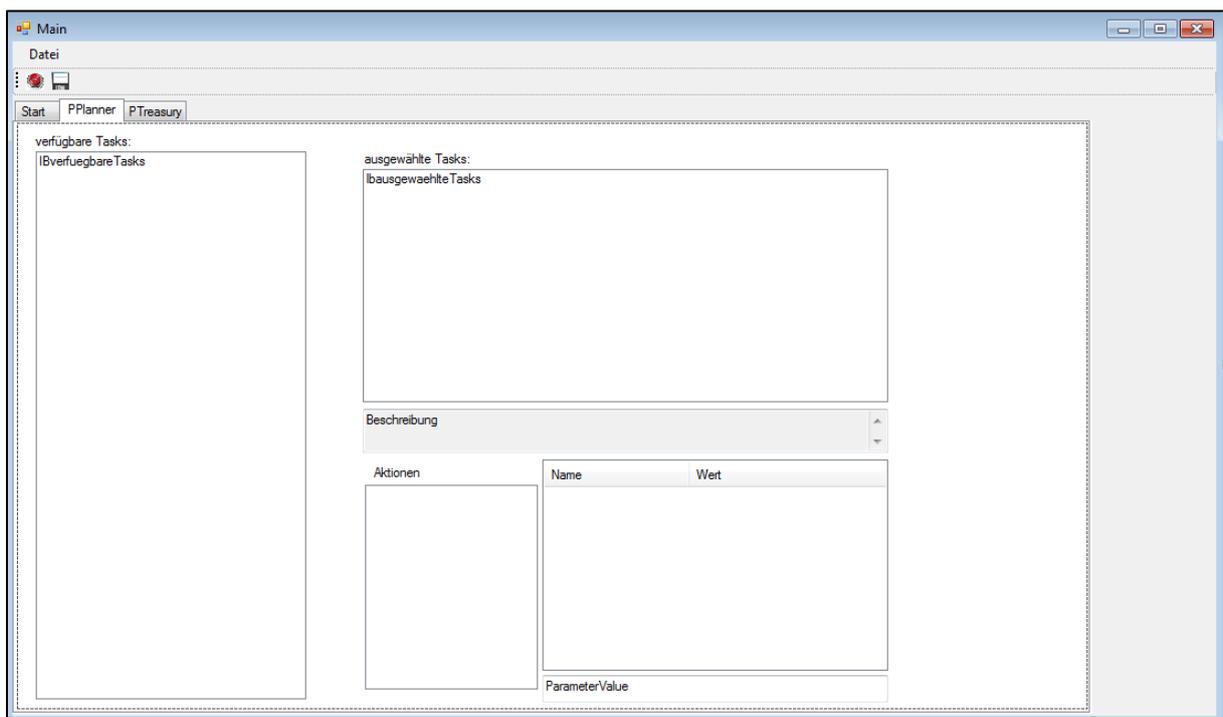


Abbildung 11: Prototype TestSuite<sup>95</sup>

Der erste Prototyp diente zur Abklärung des Oberflächendesigns. Wie Abbildung 11 zeigt, sollen alle verfügbaren Arbeitsschritte (Tasks) auf der linken Seite angezeigt werden. Ein weiteres Anzeigefenster gibt einen Überblick über die bereits ausgewählten Tasks. Da ein

<sup>92</sup> Vgl. BÖHM, R.; FUCHS, E.; PACHER, G. (1996), S. 105

<sup>93</sup> Vgl. HEJLSBERG, A.; WILTAMUTH, S.; GOLDE, P. (2007), S. 25

<sup>94</sup> Vgl. MÖSSENBÖCK, H. (2003), S. 1

<sup>95</sup> Quelle: Eigene Darstellung

Task aus mehreren einzelnen Aktionen besteht, werden diese ebenfalls in der unteren Bildschirmhälfte angezeigt. Eingabedateien bzw. Parameter werden in diesem Eingabe- und Anzeigefeld vom Benutzer definiert. Eine Möglichkeit die Konfiguration zu speichern sowie den Test zu starten wird über die Menüleiste realisiert.

Im weiteren Projektfortschritt wurden diese Prototypen permanent verfeinert und an die Anforderungen der Testabteilung angepasst. Die Vorlage dieser Modelle bei Gesprächen erleichterte die Kommunikation zwischen Auftraggeber und Entwickler, da man die theoretischen Ausführungen mit konkreten Beispielen stützen konnte.

Durch das beschriebene Vorgehen konnte das Oberflächendesign in relativ kurzer Zeit festgelegt werden. Nachdem man sich auf das Aussehen der Programmoberfläche geeinigt hatte, wurden die Eingabefelder und Schaltflächen mit der entsprechenden Funktionalität belegt. Dafür wurde die Definition des Frameworks und des internen Programmablaufes erarbeitet.

### **8.3. Die Framework Definition**

Wie im voranstehenden Kapitel erläutert, dienten Prototypen über die gesamte Entwicklungszeit als Kommunikationsgegenstand für das Oberflächendesign zwischen Entwickler und Auftraggeber. Nachdem das grundlegende Design definiert und festgelegt wurde, war die Definition der Funktionsweise der nächste Schritt im Projekt.

Nachdem die ersten Klassen für die Prototypen deklariert wurden, analysierte man mögliche Programmabläufe, um ein geeignetes Konstrukt für das Framework zu erarbeiten.

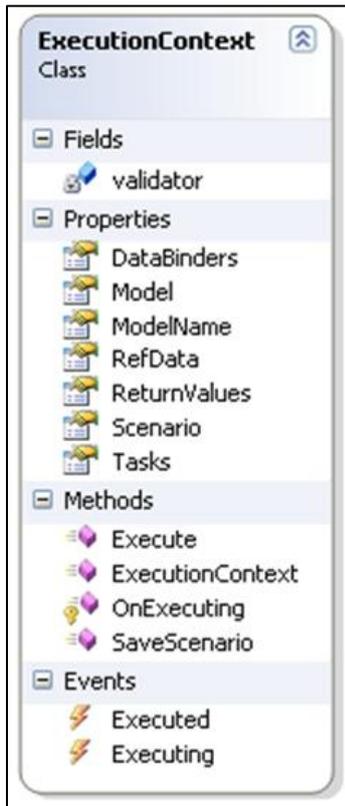


Abbildung 12: ExecutionContext<sup>96</sup>

Für die Abarbeitung der einzelnen Tasks wurde eine dafür zuständige Klasse namens ExecutionContext, wie Abbildung 12 zeigt, eingeführt. Diese Klasse beinhaltet nicht nur die verfügbaren Tasks, sondern gibt auch über die Eigenschaft „Model“ an, um welchen ExecutionContext es sich handelt. Hintergedanke dabei war, dass für jedes zu testende Programm ein eigener Context angelegt und mit den dafür notwendigen Parametern ausgestattet wird. Auch die zusammengestellten Szenarien, welche aus aneinandergereihten Tasks bestehen, werden in diesem Objekt gespeichert. Die Referenzdaten werden ebenso im Context abgelegt wie auch die Rückgabewerte der darin enthaltenen Tasks.

Die allgemeine Task-Klasse definiert Eigenschaften und Methoden, welche für alle Tasks Bedeutung haben. So hat jeder Task einen Namen, eine Beschreibung, Referenzdaten und enthält eine Liste mit den abzuarbeitenden Aktionen, wie Abbildung 13 zeigt. Von dieser Task-Klasse leiten sich die konkreten Tasks für das zu testende Programm ab. Die Task-Klasse für den Professional Planner beinhaltet alle Methoden mittels denen man das Programm steuern kann. Diese Methoden werden als Aktionen aufgerufen und ausgeführt. Ebenfalls benötigt die Task-Klasse des Professional Planners spezielle Attribute und Objekte zur Abarbeitung der Aktionen, welche in der konkreten Klasse definiert sind.

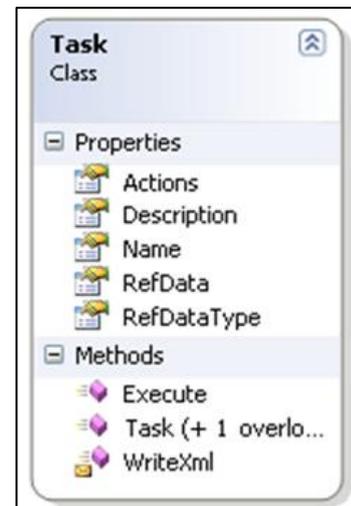


Abbildung 13: Task<sup>97</sup>

<sup>96</sup> Quelle: Eigene Darstellung

<sup>97</sup> Quelle: Eigene Darstellung



Abbildung 14: XmlTaskReader<sup>98</sup>

Die Definitionen der Tasks werden in XML-Dateien gespeichert. Darin wird festgelegt für welches Programm die Tasks verwendbar sind. Die zum Task gehörenden Aktionen (Arbeitsschritte) sowie mögliche Referenzdaten werden ebenfalls in der XML-Datei gespeichert. Abbildung 14 zeigt das Klassen-Diagramm für den XmlTaskReader. Diese Klasse liest die Daten aus der XML-Datei ein und erstellt aufgrund der eingelesenen Daten die entsprechenden Task-Objekte. Auch Informationen über den ExecutionContext werden von der XML-Datei eingelesen und weiterverarbeitet.

Aufgrund der Definition des Frameworks konnte die Implementierung effizient und schnell gestaltet werden. Die Definitionen der einzelnen Klassen und deren Zusammenwirken im Programm waren nach kurzer Entwicklungszeit fertig implementiert. Dies war jedoch nicht gleichzusetzen mit einem fertigen Programm, da laufend Erweiterungen und Anpassungen nach Besprechungen mit der Entwicklungs- sowie Testabteilung nötig waren. Wichtig war es, kleine Projektfortschritte immer wieder zu überprüfen und diese mit dem Kooperationspartner zu besprechen.

#### 8.4. Realisierung und Überprüfung der Zwischenergebnisse

Da es von Beginn an kein vollständiges Pflichtenheft der zu erstellenden Software gab, war die laufende Überprüfung des Projektfortschrittes bzw. der Zwischenergebnissen enorm wichtig. Im Idealfall existieren genau definierte Anforderungsdokumente an die man sich während der Entwicklungsphase halten muss. Im vorliegenden Fall wurden, wie Kapitel 8.1. zeigt, grobe Ziele vorgegeben. Da von Projektbeginn an nicht genau klar war, wohin sich das

---

<sup>98</sup> Quelle: Eigene Darstellung

Programm entwickeln würde, waren permanente Feedback-Runden mit allen Beteiligten sehr wichtig.

Grundlegende Designentscheidungen wurden umgehend kommuniziert und bewertet. Diese ständige Kommunikation zwischen Auftraggeber und Entwickler ist ein Hauptbestandteil des gewählten Entwicklungsprozesses. Im vorliegenden Fall wäre es nicht denkbar gewesen, vier Monate auf sich alleine gestellt zu entwickeln und abschließend das Programm zu übergeben.

Durch ständiges Feedback und Kommunikation ist man nie Gefahr gelaufen, Anforderungen falsch zu interpretieren bzw. falsch umzusetzen. Dabei war natürlich die Integration in das Testteam von der Firma WINTERHELLER software GmbH enorm hilfreich. Dadurch konnte man sehr schnell und unbürokratisch Meinungen einholen und Vorschläge sofort umsetzen.

Ständige Softwaretests sollten Fehler in der Entwicklung ausschließen. Zusätzlich war durch permanentes Testen gewährleistet, dass am Ende der Implementierungsphase die Testzeit der fertigen Software sehr kurz gehalten werden konnte.

## **8.5. Testen der Software**

Benutzt eine Testabteilung eine Software zum automatisierten Testen des in der Unternehmung entwickelten Programms, muss davon ausgegangen werden, dass dieses Testtool fehlerfrei funktioniert. Ansonsten wäre es nicht nachvollziehbar, ob ein etwaiger dokumentierter Fehler des Testtools wirklich in dem zu testenden Programm liegt oder das Testprogramm selbst fehlerhaft gearbeitet hat.

Daher wurden während der Entwicklung der WINTERHELLER TestSuite fortlaufend die Zwischenresultate getestet. Auch, wenn alle Zwischentests erfolgreich absolviert wurden, kann nicht von einer Fehlerfreiheit des Endprodukts ausgegangen werden. Überhaupt kann die Abwesenheit von Fehlern nicht die Korrektheit eines Programmes gewährleisten, wie in Kapitel 5 beschrieben.

Für den Test am fertigen Programm wurde ein Testszenario, welches in Kapitel 9 vorgestellt wird, entwickelt. Manche Ergebnisse dieses Testszenarios wurden per Hand errechnet. Andere sind mittels einer Version vom Professional Planner erstellt worden, die schon seit mehreren Monaten ohne Fehlfunktionen bei Kunden der Firma WINTERHELLER software GmbH im Einsatz waren. Diese Ergebnisse dienten als Referenzdaten und wurden für die Verifizierung der Korrektheit des Testtools herangezogen. Das erstellte Szenario wurde mittels der WINTERHELLER TestSuite abgearbeitet und Referenzdaten mit den tatsächlichen Ergebnissen verglichen.

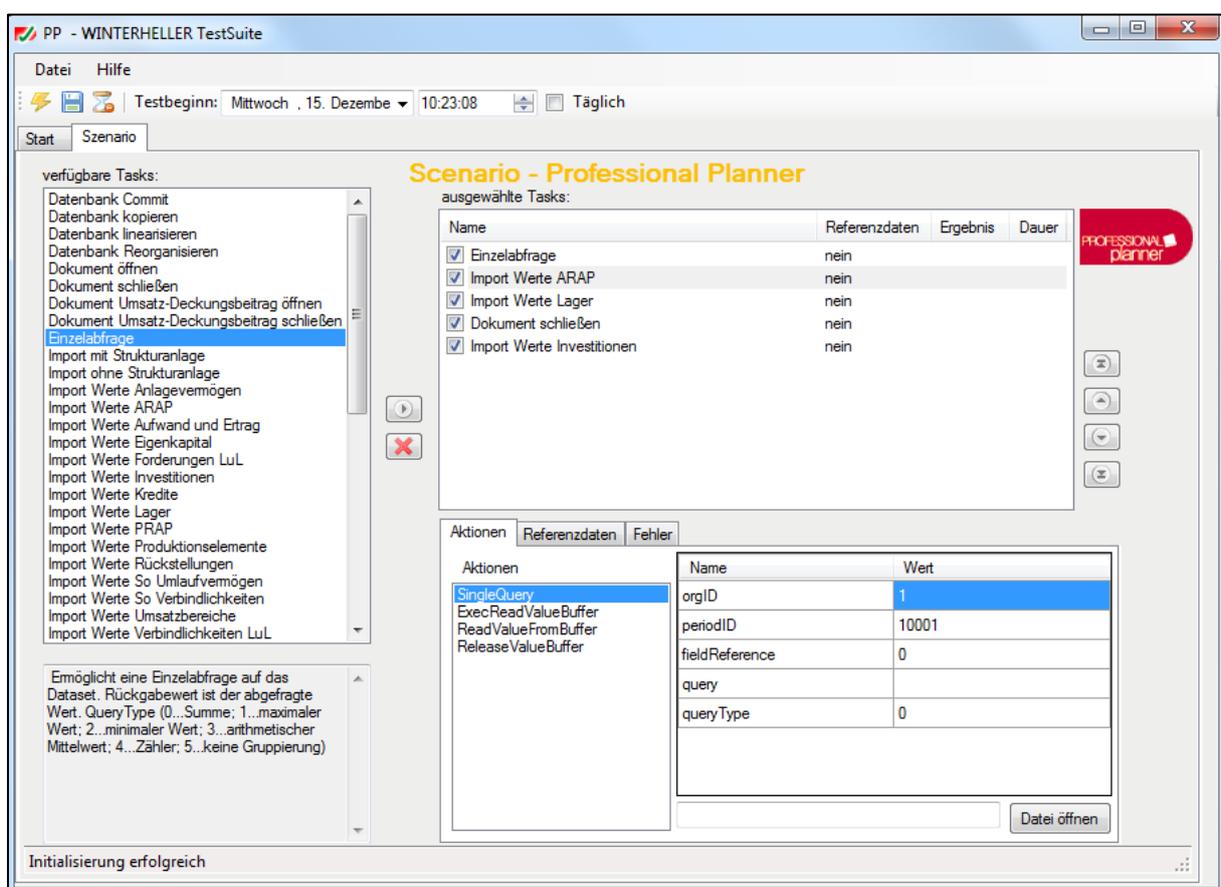


Abbildung 15: WINTERHELLER TestSuite<sup>99</sup>

Die fertige TestSuite, siehe Abbildung 15, wurde mit verifizierten bzw. überprüften Daten getestet. Etwaige Abweichungen zwischen den Ergebnissen und Referenzdaten konnten somit auf Fehlberechnungen in der TestSuite zurückgeführt werden. Zu erwähnen ist, dass hier speziell die Validierungsfunktionen im Testtool unter enormer Beobachtung standen.

<sup>99</sup> Quelle: Eigene Darstellung

Diese Funktionen überprüfen die Ergebnisse aus dem Professional Planner mit den Vorgabewerten. Dabei ist es von extrem hoher Wichtigkeit, dass diese Validierungsfunktionen korrekt die Werte vergleichen. Die Ergebnisse dieser Validierungsfunktionen werden in ein Testlog gespeichert und dem Benutzer nach Testende angezeigt. Würden diese Mechanismen nicht korrekt arbeiten, könnte der Benutzer nicht verifizieren, ob es sich um einen Fehler der TestSuite oder des zu testenden Programmes handelt.

## 9. Testscenario

Um mittels der WINTERHELLER TestSuite den Professional Planner testen zu können, werden Eingabedaten und entsprechende Ausgabedaten bzw. Referenzdaten benötigt. Aus diesem Grund war neben der Erstellung einer Testsoftware auch die Generierung geeigneter Testdaten Bestandteil der Diplomarbeit. Das Kapitel 9 gibt einen Einblick in die Art und den Aufbau dieser Testdaten, welche zu einem Testscenario zusammengestellt wurden. In Anlehnung auf das bereits in Kapitel 7 erwähnte Standardfallbeispiel wurde ebenfalls ein fiktives Fallbeispiel konzipiert. Die gesamten Angaben für das Testscenario befinden sich im Anhang.

### 9.1. Fallbeispiel Evolution Production GmbH

Die Evolution Production GmbH wurde 1980 von den drei Gesellschaftern Reinfried Gruber, Horst Mengele und Niklas Bergström in Wien gegründet. Zu Beginn konzentrierte sich das Unternehmen auf die Zulieferung von Elektronikbauteilen für Fahrzeughersteller. Anfang der Neunziger wurde Graz als zweiter Produktionsstandort aufgebaut. Das neue Werk in Graz war auf die Produktion eines selbstentwickelten Autoradios ausgerichtet. Mitte der Neunziger wurden Sicherheitssysteme in die Produktpalette aufgenommen. Am Anfang des neuen Jahrtausends rüstete man die Werke in Wien für die Produktion von hochwertigen Navigationsgeräten, welche auch im Einzelhandel angeboten werden, um. 2005 erweiterte man die Firma um einen Entwicklungs- und Servicestandort in Salzburg. In den darauffolgenden drei Jahren stieg das Umsatzvolumen stetig an bis man im Jahr 2008 die 400 Millionen Umsatzgrenze erreichte.

Das Jahr 2009 stellte für die Fahrzeughersteller sowie für die Zulieferindustrie ein äußerst turbulentes Jahr dar. Auch die Evolution Production GmbH war von drastischen Nachfrageeinbrüchen betroffen.

Damit in wirtschaftlich schweren Zeiten wichtige Entscheidungen so früh wie möglich getroffen werden können, führte die Controlling-Abteilung eine Planungssoftware ein. Der Abteilungsleiter entschied sich nach einigen Demonstrationen für das Produkt Professional Planner der Firma Winterheller.

Die folgenden Angaben dienen der Eingabe im Professional Planner, damit die Controlling-Abteilung die gewünschten Plandaten erheben kann. Dafür ist eine Aufbereitung und anschließende Überführung der Datensätze in die Software nötig. Informationen in Fußnoten für Einstellungsmöglichkeiten beziehen sich auf die Bedienung des Professional Planners.

## **9.2. Erfolgsplanung**

Die zu planenden Ein- und Auszahlungen gehen in die Finanzplanung ein und werden dort mit den Finanzierungsmöglichkeiten abgestimmt. Den Abschluss der kurzfristigen Planung bildet die Erfolgsplanung. Dabei werden die geplanten Absatzmengen mit den geplanten proportionalen Selbstkosten der Plankalkulation bewertet. Als Ergebnis erhält man die proportionalen Selbstkosten des Planumsatzes. Subtrahiert man diese von den Planerlösen, erhält man den Plan-Deckungsbeitrag. Nach Abzug der fixen Kosten errechnet sich der Plan-Erfolg.<sup>100</sup>

Für das Fallbeispiel Evolution Production GmbH wurden Umsatzdaten, variable Kosten, fixe Kosten und Verwaltungskosten für drei Standorte geplant.

---

<sup>100</sup> Vgl. KILGER, W.; PAMPEL, J.; VIKAS, K. (2007), S. 655

Beispielhaft zeigt die nachstehende Tabelle die Umsatzdaten und variablen Kosten für den Standort Wien.

	Navi GlobalConnection	Navi EasyWay	Navi DWF <sup>101</sup>
Absatzmenge Stk	176.549	307.570	132.514
Nettopreis	240,00 €	150,00 €	410,00 €
Umsatzsteuer	20 %	20 %	20 %
Var. Kosten (WES)	50 %	52 %	229,60 €/Stk
Rabatte	1 %	2,5 €/Stk	2 %
Skonto	856.082,0 €	1,5 %	869.292,0 €
So variable Kosten	3 %	2,4 €/Stk	3,2 €/Stk
Vertriebskosten	2,1 €/Stk	1,2 €/Stk	6 %

Tabelle 4: Plandaten Wien<sup>102</sup>

Der Professional Planner bietet die Möglichkeit, die Umsatzverteilung der Produkte in absoluten Mengen oder einer Prozentverteilung für das Planjahr anzugeben. Tabelle 5 zeigt die Umsatzverteilung des Produktes Navi GlobalConnection für das zu planende Jahr.

Navi GlobalConnection:

Jan.	Feb.	März	April	Mai	Juni
12.138	11.034	12.138	13.241	13.241	15.448
Juli	Aug.	Sept.	Okt.	Nov.	Dez.
16.551	13.241	15.448	16.551	17.655	19.862

Tabelle 5: Produktionsmengen GlobalConnection<sup>103</sup>

<sup>101</sup> DWF = Drive Walk Fly

<sup>102</sup> Quelle: Eigene Darstellung

<sup>103</sup> Quelle: Eigene Darstellung

### 9.3. Produktionsplanung

Damit die Produkte oder Dienstleistungen am Markt angeboten werden können, müssen diese in entsprechender Menge und Qualität produziert werden. Die einzelnen Schritte von der Beschaffung über die Produktion bis hin zum Absatz müssen genau aufeinander abgestimmt werden. Steigende Absatzzahlen sind durch eine Ausweitung der Produktionskapazitäten in der Planung zu berücksichtigen.<sup>104</sup>

Im vorliegenden Fallbeispiel wird von zwei Arten von Lagern ausgegangen:

- Teilelager
- Produktionslager

Auf dem Teilelager sollen die zugekauften Elektronikbauteile gebucht werden. Die jeweils produzierten Einheiten werden dem Produktionslager zugeschrieben.

Der Professional Planner bietet für die Erfassung der Produktionsdaten verschiedene Eingabemöglichkeiten an. In Tabelle 6 werden die Daten zur Erfassung der Produktionsplanung des Standortes Wien für das Fallbeispiel gezeigt.

	Navi GlobalConnection	Navi EasyWay	Navi DWF
Produktionsmenge	176.549	307.570	132.514
Direkte Kosten	40,00 €/Einheit	38,00 €/Einheit	110,00 €/Einheit
Einkaufspreis Elektronikmodul	73,00 €	35,00 €	109,00 €
Vorsteuer	20 %	20 %	20 %
Gemeinkosten/ Elektronikmodul	10 %	14 %	10 %

Tabelle 6: Produktionsplanung Wien<sup>105</sup>

<sup>104</sup> Vgl. MÖRTL, J.; PINTER, G.; SPONRING, C. (2009), S. 100f.

<sup>105</sup> Quelle: Eigene Darstellung

Zur Vereinfachung wird angenommen, dass die jeweiligen Produktionsmengen über das Planjahr gleichverteilt werden. Pro produzierte Einheit wird ein Elektronikmodul verbaut, welches über einen Lieferanten zugekauft wird.

#### **9.4. Finanzplanung und Planbilanz**

Der Finanzplan enthält die erwarteten Zahlungsüberschüsse bzw. den Zahlungsbedarf für die Planperiode. Ebenfalls fließen Informationen über das Investitionsbudget als wertmäßige Zusammenstellung der geplanten Investitionsvorgaben in den Finanzplan ein. Auch die zeitlichen Verteilungen von Ein- bzw. Auszahlungen und die voraussichtlich verfügbaren Finanzierungsquellen sind Bestandteile des Finanzplanes. Abschließend lassen sich die Gesamtergebnisse des Budgetierungsprozesses in der Planbilanz darstellen.<sup>106</sup>

Das Fallbeispiel enthält unter diesem Punkt Angaben zu Konten welche für die Erstellung der Finanzplanung bzw. Planbilanz nötig sind. Darunter fallen Angaben zu den Sachanlagen, Finanzanlage, Verbindlichkeiten sowie Rückstellungen und einige mehr.

Zum Beispiel werden die Anfangsbestände der passiven Rechnungsabgrenzungen sowie deren partiellen Auflösungen im Laufe des Planjahres angegeben.

#### **9.5. Investitionen**

Der Investitionsprozess kann in die Planungsphase, die Realisations- bzw. Erstellungsphase sowie die Nutzungs- bzw. Betriebsphase unterschieden werden. Die Entscheidung über die Durchführung einer Investition ist der Planungsphase zuzuordnen.<sup>107</sup>

Für das Fallbeispiel sind zwei große Investitionen zu planen. Am Standort Wien wird eine neue Produktionsmaschine und am Standort Graz eine Verpackungsmaschine angeschafft.

---

<sup>106</sup> Vgl. EWERT, R.; WAGENHOFER, A. (1993), S. 418

<sup>107</sup> Vgl. GÖTZE, U. (2008), S. 14

Für die beiden Anschaffungen werden die buchhalterischen sowie kalkulatorischen Abschreibungen berücksichtigt. Auch die Finanzierungsform (Kredit) fließt in das Testszenario ein.

## 9.6. Steuern

In die Planung sollen nur solche Steuern einfließen deren Kostencharakter unumstritten ist. Da die Steuern meist jährlich oder vierteljährlich gezahlt werden, ergibt sich ein Abgrenzungsproblem bei kurzen Abrechnungsperioden.<sup>108</sup>

Im Testszenario wird nur die Körperschaftsteuer berücksichtigt bzw. Steuervorauszahlungen finden keine statt.

## 9.7. Referenzdaten

Anhand der Referenzdaten kann überprüft werden ob das Programm richtig funktioniert bzw. die integrierte Rechenlogik fehlerfrei abgebildet wurde. Dazu werden die definierten Eingabedaten in der TestSuite eingelesen. Die Testsoftware gibt die aufbereiteten Daten an das zu testende Programm, dem Professional Planner, wie Abbildung 16 zeigt weiter. Durch die Eingabe werden im Professional Planner Berechnungen durchgeführt und deren Ergebnisse von der TestSuite ausgelesen. Die TestSuite validiert diese Werte mit den vom Benutzer angegebenen Referenzdaten. Die Testsoftware protokolliert die Testergebnisse und etwaige Abweichungen in einem Testlog. Diese Testdokumentation wird mit einem Zeitstempel abgespeichert um für spätere Testauswertungen verfügbar zu sein.

---

<sup>108</sup> Vgl. JOST, H. (1996), S. 69f.

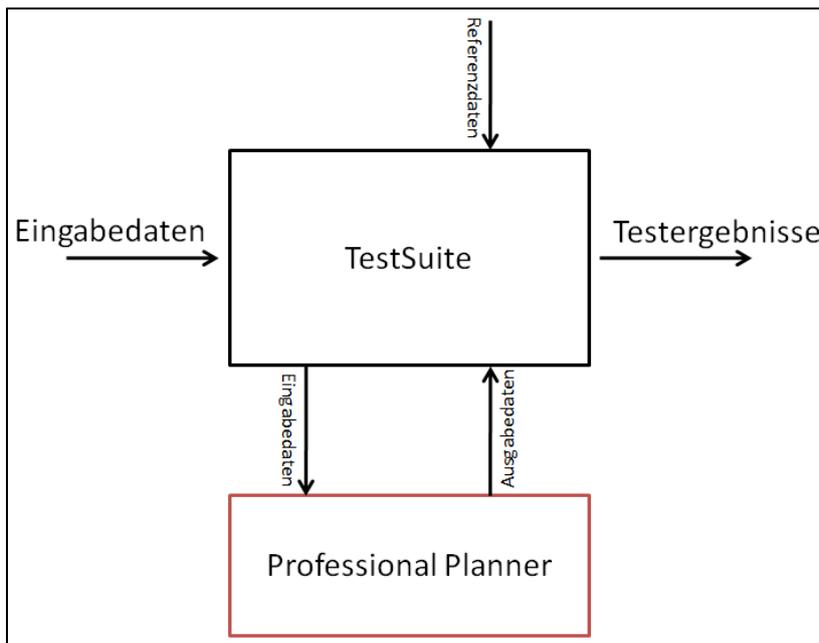


Abbildung 16: Testablauf<sup>109</sup>

Da bei manchen Berechnungen Abweichungen in der dritten oder vierten Nachkommastelle vernachlässigbar sind, hat der Benutzer die Möglichkeit Toleranzen in der TestSuite anzugeben.

Manche Referenzdaten wurden per Hand errechnet und andere mittels einer validierten und freigegebenen Version des Professional Planners generiert.

Damit die WINTERHELLER TestSuite die Referenzdaten lesen und verarbeiten kann müssen diese in eine bestimmte Form gebracht werden. Die Eingabewerte werden im Professional Planner über drei Dimensionen erfasst:

- OrgID
- Zeit-ID
- Feldbezug

---

<sup>109</sup> Quelle: Eigene Darstellung

Die Daten müssen daher wie folgt in Form einer CSV-Datei oder Excel-Datei aufbereitet werden:

OrgID	Zeit-ID	Feldbezug	Wert
8	10001	101	5462450
9	10001	101	52388130

**Tabelle 7: Referenzdaten I**<sup>110</sup>

Zur besseren Lesbarkeit und Verständlichkeit können zusätzliche Spalten in die Datensätze aufgenommen werden:

OrgID	Name	Zeit-ID	Feldbezug	FB-Beschreibung	Wert
8	GreenRadio SF	10001	101	Nettoerlöse	5462450
9	GreenRadio FF	10001	101	Nettoerlöse	52388130

**Tabelle 8: Referenzdaten II**<sup>111</sup>

Jedes eingefügte Strukturelement ist eindeutig über eine Organisations-ID (OrgID) identifizierbar. Neben diesem Identifier ist auch die Bezeichnung (Name) ein eindeutiges Unterscheidungsmerkmal. Die Zeit-ID gibt an für welchen Zeitraum der angegebene Wert gültig ist. In Tabelle 8 bedeutet die Zeit-ID 10001, dass der Wert für das gesamte Jahr steht.

---

<sup>110</sup> Quelle: Eigene Darstellung

<sup>111</sup> Quelle: Eigene Darstellung

## **10. Potential der Testsoftware**

Kürzere Entwicklungszeiten und eine stetige Verbesserung der Qualität bei Softwareprojekten machen effektives Testen unabdingbar. Mechanismen oder wie in diesem Fall Testtools erleichtern die Arbeit einer Testabteilung bzw. gestalten den Testvorgang effizienter da in kürzerer Zeit größere Datenmengen getestet werden können.

### **10.1. Vorteile durch die Verwendung der Testsoftware**

Durch den Einsatz der WINTERHELLER TestSuite ist es der Testabteilung möglich in kürzerer Zeit größere Testdatenmengen zu evaluieren. Waren in der Vergangenheit zwei Mitarbeiter über zwei bis drei Tage damit beschäftigt die Standarddatensätze in eine neue Version einzuspielen und Ergebnisse oder Einstellungen auf deren Korrektheit zu prüfen, benötigt dafür die WINTERHELLER TestSuite wenige Minuten.

Die zeitliche Reduktion der Testzeit bedeutet nicht gleichfalls ein dadurch entstandenes Einsparungspotential an Arbeitskräften. Die Tester können an anderer Stelle eingesetzt werden. So sind Oberflächentests, die auf das Design und auf die korrekte Darstellung der Programmoberfläche abzielen, noch immer manuell durchzuführen. Die eingesparte Zeit kann in weiterführende Tests bzw. den Ausbau der Testdokumentation investiert werden. Dadurch wird die Qualität der Testabläufe und in der Endkonsequenz die Qualität der erstellten Programme gesteigert.

### **10.2. Risiken durch die Verwendung der Testsoftware**

Wo Menschen arbeiten können Fehler passieren. Herunter gebrochen auf die WINTERHELLER TestSuite bedeutet dies, dass auch hier zwei entscheidende Fehlerpotentiale existieren. Einerseits birgt ein fehlerhaftes Testprogramm Gefahren für einen korrekten

Testablauf, andererseits müssen Testdaten sowie die dazugehörigen Referenzdaten fehlerlos und lückenlos sein um aussagekräftige Testergebnisse zu generieren.

Werden Szenarien durchgespielt, die nicht den Funktionsumfang der Programme abdecken, kann keine ausreichende Aussage über die Anwesenheit bzw. Abwesenheit von Fehlern gemacht werden. Den Anspruch auf Vollständigkeit wird in der Regel kein Test erheben können, da dies bei derartig komplexen Programmen nicht möglich ist. Es muss jedoch genügend Funktionalität getestet werden um einen hohen Prozentsatz des Leistungsumfanges abdecken zu können.

Am Ende steht als Risikopotential natürlich der Faktor Mensch als Benutzer der Software. Die falsche Verwendung der TestSuite kann zu Fehlern bzw. Nichterkennung von diesen führen. Wie bereits erörtert, können Fehlertoleranzen eingestellt werden. Sind diese Einstellungen zu großzügig gewählt, ist es möglich, dass Fehler als Rundungsdifferenzen abgetan werden.

Der Erfolg des Einsatzes automatisierter Tests ist wesentlich vom Benutzer abhängig.

### **10.3. Anwendungspotential der Testsoftware**

Im Zuge der Entwicklung der WINTERHELLER TestSuite wurde besonderes Augenmerk auf die Erweiterbarkeit der Software gesetzt. Ein Ziel war es, in Zukunft die Testsoftware auch für die beiden anderen Softwarelösungen Professional Consolidation und Professional Treasury einsetzen zu können. Dafür wurde das Design entsprechend offen gehalten und im Programmcode selbst wurden entsprechende Rumpfklassen, jedoch noch ohne Funktionalität, implementiert. Um die angesprochenen Softwarelösungen testen zu können, müssen wie auch für den Professional Planner entsprechende Testszenarien konzipiert werden.

## 11. Zusammenfassung und Resümee

Planung dient der Umsetzung zukünftiger Ziele. Diese Ziele zeigen was ein Unternehmen erreichen will. Unternehmensplanung kann in wirtschaftlich schwierigen Zeiten Stabilität für eine Unternehmung bedeuten. Das Durchspielen verschiedener Handlungsalternativen im Zuge der Planungsphase bereitet Unternehmen bestmöglich auf die Zukunft und mögliche Krisen vor.

Unternehmen kommen bei der Bewältigung dieser Planungsaufgaben kaum noch ohne ein elektronische Datenverarbeitung aus. Heutzutage gibt es immer umfangreichere Programme die dem Benutzer im Zuge des Planungsvorganges unterstützen.

Der Kooperationspartner dieser Diplomarbeit, WINTERHELLER software GmbH, bietet mit dem Professional Planner eine Planungs- und Budgetierungslösung für jede Unternehmensgröße an. Diese Softwarelösung ermöglicht es dem Benutzer Auswertungen und Vergleiche auf Basis betriebswirtschaftlicher Planungsrechnung zu erstellen. Damit ist es möglich verschiedenartige Szenarien durchzuspielen und zu bewerten.

Aufgrund der zunehmenden Komplexität dieser Planungssoftware wird es schwieriger Fehlerfreiheit im Zuge des Software-Entwicklungsprozesses zu garantieren. Daher wird die Arbeit der Testabteilung immer wichtiger. Sie ist im Endeffekt für die Freigabe der fertiggestellten Programme verantwortlich. Die Mitarbeiter haben die Aufgabe, die Funktionalität der Softwarelösungen zu testen und damit sicherzustellen, dass Kunden ein fehlerfreies Programm erwerben. Wegen der angesprochenen Komplexität werden Tests immer schwieriger bzw. langwieriger. Automatische Softwaretests ermöglichen der Testabteilung bei steigender Funktionalität schneller und effizienter testen zu können.

## Abkürzungsverzeichnis

BCL	Business Content Library
BI	Business Intelligence
CPM	Corporate Performance Management
CSV	Comma-Separated Values
EDV	Elektronische Datenverarbeitung
GuV	Gewinn- und Verlustrechnung
OLAP	Online Analytical Processing
OLCAP	Online Calculating and Analytical Processing
PP	Professional Planner
SQL	Structured Query Language
XML	Extensible Markup Language
XP	Extreme Programming

## Abbildungs- und Tabellenverzeichnis

Abbildung 1: Professional Planner .....	6
Abbildung 2: OLCAP-Würfel .....	8
Abbildung 3: Ungenutztes Leistungspotenzial .....	14
Abbildung 4: Szenarien.....	18
Abbildung 5: Unternehmensbudget.....	21
Abbildung 6: Black-Box-Test.....	32
Abbildung 7: Wasserfallmodell .....	38
Abbildung 8: V-Modell.....	38
Abbildung 9: Testablauf .....	42
Abbildung 10: Taskmodell .....	47
Abbildung 11: Prototype TestSuite .....	48
Abbildung 12: ExecutionContext.....	50
Abbildung 13: Task .....	50
Abbildung 14: XmlTaskReader .....	51
Abbildung 15: WINTERHELLER TestSuite .....	53
Tabelle 1: Unternehmensergebnis.....	22
Tabelle 2: Finanzplan.....	23
Tabelle 3: Einsparungen durch Test-Automatisierung.....	34
Tabelle 4: Plandaten Wien .....	57
Tabelle 5: Produktionsmengen GlobalConnection .....	57
Tabelle 6: Produktionsplanung Wien .....	58
Tabelle 7: Referenzdaten I .....	62
Tabelle 8: Referenzdaten II .....	62

# Literaturverzeichnis

## Bücher

BAIER, P. (2007): Praxishandbuch Controlling – Unternehmensplanung und Reporting; München 2007

BÖHM, R.; FUCHS, E.; PACHER, G. (1996): System-Entwicklung in der Wirtschaftsinformatik; 4. Auflage; Zürich 1996

BUNSE, C.; KNETHEN, A. (2008): Vorgehensmodelle kompakt; 2. Auflage; Heidelberg 2008

DUSTIN, E.; RASHKA, J.; PAUL, J. (2000): Software automatisch testen – Verfahren, Handhabung und Leistung; Berlin 2000

EGGER, A.; WINTERHELLER, M. (2004): Kurzfristige Unternehmensplanung – Budgetierung; 13. Auflage; Wien 2004

EHRMANN, H. (2007): Kompendium der praktischen Betriebswirtschaft: Unternehmensplanung; 5. überarbeitete und aktualisierte Auflage; Leipzig 2007

EISL, C.; HANGL, C.; LOSBICHLER, H.; MAYR, A. (2008): Grundlagen der finanziellen Unternehmensführung; Wien 2008

ERICHSEN, J. (2010): Betriebswirtschaftliche Grundlagen; Freiburg 2010

EWERT, R.; WAGENHOFER, A. (1993): Interne Unternehmensrechnung; Berlin 1993

FINK, C. (2003): Prozessorientierte Unternehmensplanung – Analyse, Konzeption und Praxisbeispiele; Wiesbaden 2003

FISCHER, J. (1989): Qualitative Ziele in der Unternehmensplanung; Berlin 1989

FRÜHAUF, K.; LUDEWIG, J.; SANDMAYR, H. (2006): Software-Prüfung – Eine Anleitung zum Test und zur Inspektion; Zürich 2006

GLEICH, R.; KLEIN, A. (2009): Moderne Budgetierung; Freiburg 2009

GÖTZE, U. (2008): Investitionsrechnung – Modelle und Analyse zur Beurteilung von Investitionsvorhaben; 6. Auflage; Berlin 2008

HANSMANN, F. (1990): Quantitative Betriebswirtschaftslehre – Lehrbuch der modellgestützten Unternehmensplanung; 3. überarbeitete Auflage; München/Wien 1990

HEJLSBERG, A.; WILTAMUTH, S.; GOLDE, P. (2007): Die C#-Programmiersprache; 2. Auflage; München 2007

HORNGREN, C.T.; FOSTER, G.; DATAR, S.M. (2000): Cost accounting – a managerial emphasis; 10<sup>th</sup> edition; New York 2000

JOST, H. (1996): Kosten- und Leistungsrechnung; 7. Auflage; Wiesbaden 1996

KILGER, W.; PAMPEL, J.; VIKAS, K. (2007): Flexible Plankostenrechnung und Deckungsbeitragsrechnung; 12. Auflage; Wiesbaden 2007

KRALICEK, P. (1998): Planbilanzen; Wien/Frankfurt 1998

KRETSCHMER, P. (1979): Unternehmensplanung; München 1979

KRYSTEK, U.; MOLDENHAUER, R. (2007): Handbuch Krisen- und Restrukturierungsmanagement; Stuttgart 2007

LIGGESMEYER, P. (2009): Software Qualität – Testen, Analysieren und Verifizieren von Software; 2. Auflage; Heidelberg 2009

MÖRTL, J.; PINTER, G.; SPONRING, C. (2009): Erfolgsfaktor Unternehmensplanung – Grundlagen und Praxistipps; 1. Auflage; Wien 2009

MÖSSENBÖCK, H. (2003): Softwareentwicklung mit C# ; Heidelberg 2003

MYERS, G.J. (1989): Methodisches Testen von Programmen; Oldenbourg 1989

PARRINGTON, N.; PARRINGTON, M. (1991): Software Test – Ziele, Anwendungen, Methoden; Hamburg 1991

RIEG, R. (2008): Planung und Budgetierung; Wiesbaden 2008

SCHATTEN, A.; BIFFL, S.; DEMOLSKY, M.; GOSTISCHA-FRANTA, E.; ÖSTREICHER, T.; WINKLER, D. (2010): Best Practice Software-Engineering; Heidelberg 2010

SNEED, H.; BAUMGARTNER, M.; SEIDL, R. (2007): Der Systemtest – Anforderungsbasiertes Testen von Software-Systemen; München/Wien 2007

SPILLNER, A.; LINZ, T. (2010): Basiswissen Softwaretest; Heidelberg 2010

STARK, P. (2006): Das 1 x 1 des Budgetierens; Weinheim 2006

STEINER, G.A. (1969): Sprache und Schweigen, aus dem Englischen von Axel Kaun; London 1969

THALLER, G.E. (2002): Software-Test – Verifikation und Validation; Hannover 2002

VETTER, M. (1988): Strategie der Anwendungsentwicklung – Planung, Prinzipien, Konzepte; Stuttgart 1988

VIGENSCHOW, U. (2005): Objektorientiertes Testen und Testautomatisierung in der Praxis – Konzepte, Techniken und Verfahren; Heidelberg 2005

VOSSBEIN, R. (1974): Unternehmensplanung – Grundlagen und praktische Anwendung der Planung als Steuerungs-Instrument; Düsseldorf/Wien 1974

## **Zeitschriften**

TREUZ, W. (2004): Budget und Budgeting – Das klassische Planungs- und Steuerungsinstrument der KMU, in: Controller Magazin, Heft 4/04 2004, S. 340

## **Internet**

WINTERHELLER software GmbH

URL: <http://www.winterheller.com/software/home.html>, Stand: 15.10.2010, Abfrage vom: 10.12.2010

## **Sonstiges**

WINTERHELLER software GmbH

URL: [http://www.winterheller.com/pdf/whitepaper\\_olcap.pdf](http://www.winterheller.com/pdf/whitepaper_olcap.pdf), Stand: 04.01.2011, Abfrage vom: 04.01.2011

# ANHANG

## A.1. Testszenario

Die folgenden Seiten beinhalten Angaben zu dem fiktiven Testszenario „Evolution Production GmbH“.

Die Evolution Production GmbH wurde 1980 von den drei Gesellschaftern Reinfried Gruber, Horst Mengele und Niklas Bergström in Wien gegründet. Zu Beginn konzentrierte sich das Unternehmen auf die Zulieferung von Elektronikbauteilen für Fahrzeughersteller. Anfang der Neunziger wurde Graz als zweiter Produktionsstandort aufgebaut. Das neue Werk in Graz war auf die Produktion eines selbstentwickelten Autoradios ausgerichtet. Mitte der Neunziger wurden Sicherheitssysteme in die Produktpalette aufgenommen. Am Anfang des neuen Jahrtausends rüstete man die Werke in Wien für die Produktion von hochwertigen Navigationsgeräten, welche auch im Einzelhandel angeboten werden, um. 2005 erweiterte man die Firma um einen Entwicklungs- und Servicestandort in Salzburg. In den darauffolgenden drei Jahren stieg das Umsatzvolumen stetig an bis man im Jahr 2008 die 400 Millionen Umsatzgrenze erreichte.

Das Jahr 2009 stellte für die Fahrzeughersteller sowie für die Zulieferindustrie ein äußerst turbulentes Jahr dar. Auch die Evolution Production GmbH war von drastischen Nachfrageeinbrüchen betroffen.

Damit in wirtschaftlich schweren Zeiten wichtige Entscheidungen so früh wie möglich getroffen werden können führt die Controlling-Abteilung eine Planungssoftware ein. Der Abteilungsleiter entschied sich nach einigen Demonstrationen für das Produkt Professional Planner der Firma Winterheller.

Die folgenden Angaben dienen der Eingabe im Professional Planner damit die Controlling-Abteilung die gewünschten Plandaten erheben kann. Dafür ist eine Aufbereitung und anschließende Überführung der Datensätze in die Software nötig. Informationen in Fußnoten für Einstellungsmöglichkeiten beziehen sich auf die Bedienung des Professional Planners.

## A.2. Erfolgsplanung

### Umsatzdaten und variable Kosten Wien

Für den Standort Wien ist ein Profitcenter anzulegen. In Wien werden drei Arten von Navigationsgeräten produziert.

Die Controlling-Abteilung erarbeitete folgende Zahlen für das Planungsjahr:

	Navi GlobalConnection	Navi EasyWay	Navi DWF <sup>112</sup>
Absatzmenge Stk	176.549	307.570	132.514
Nettopreis	240,00 €	150,00 €	410,00 €
Umsatzsteuer	20 %	20 %	20 %
Var. Kosten (WES)	50 %	52 %	229,60 €/Stk
Rabatte	1 %	2,5 €/Stk	2 %
Skonto	856.082,0 €	1,5 %	869.292,0 €
So variable Kosten	3 %	2,4 €/Stk	3,2 €/Stk
Vertriebskosten	2,1 €/Stk	1,2 €/Stk	6 %

Tabelle 9: Plandaten Wien

	Sonst. betriebliche Erträge
Nettoumsatz	5.264.000,00 €

Tabelle 10: Sonst. betriebl. Erträge Wien

Die sonstigen betrieblichen Erträge werden über die Monate gleich.

Die Umsatzverteilung der Produkte basiert auf Erfahrungswerte aus den vorangegangenen Geschäftsjahren (in Stk):

#### Navi GlobalConnection:

---

<sup>112</sup> DWF = Drive Walk Fly

<b>Jan.</b> 12.138	<b>Feb.</b> 11.034	<b>März</b> 12.138	<b>April</b> 13.241	<b>Mai</b> 13.241	<b>Juni</b> 15.448
<b>Juli</b> 16.551	<b>Aug.</b> 13.241	<b>Sept.</b> 15.448	<b>Okt.</b> 16.551	<b>Nov.</b> 17.655	<b>Dez.</b> 19.862

Tabelle 11: Produktionsmengen GlobalConnection

**Navi EasyWay:**

<b>Jan.</b> 25.265	<b>Feb.</b> 25.265	<b>März</b> 27.462	<b>April</b> 21.969	<b>Mai</b> 23.068	<b>Juni</b> 24.166
<b>Juli</b> 25.265	<b>Aug.</b> 26.363	<b>Sept.</b> 24.166	<b>Okt.</b> 26.363	<b>Nov.</b> 28.560	<b>Dez.</b> 29.659

Tabelle 12: Produktionsmengen EasyWay

**Navi DWF:**

<b>Jan.</b> 9.939	<b>Feb.</b> 9.939	<b>März</b> 11.043	<b>April</b> 11.043	<b>Mai</b> 12.147	<b>Juni</b> 11.043
<b>Juli</b> 13.251	<b>Aug.</b> 12.147	<b>Sept.</b> 11.043	<b>Okt.</b> 9.939	<b>Nov.</b> 9.939	<b>Dez.</b> 11.043

Tabelle 13: Produktionsmengen DWF

## Fixe Kosten Wien

Bestimmte Fixkosten können direkt auf die Standorte aufgeteilt werden. Die nicht direkt zurechenbaren Fixkosten werden auf Unternehmensebene zusammengefasst. Die Planfixkosten werden aus den Vorjahresdaten abgeleitet.

Personal	37.600.000,00 €
Sonstiger betrieblicher Aufwand	14.100.000,00 €

Tabelle 14: Fixkosten Wien

### Personal:

Die Personalkosten werden auf die Monate gleichverteilt.

Einstellungen:

- Erfolgswirkung: ..... Aufwand Sonstiger
- Bilanzkonto: ..... So Verbindlichkeiten
- Detailkonto: ..... [Verbindlichkeiten Gehalt]

Sonstiger betrieblicher Aufwand:

Die sonstigen betrieblichen Aufwendungen werden ebenfalls gleichverteilt, die Vorsteuer ist mit 20 % anzusetzen.

Einstellungen:

- Erfolgswirkung: ..... Aufwand Sonstiger
- Bilanzkonto: ..... So Verbindlichkeiten
- Detailkonto: ..... [So Verbindlichkeiten]

## Umsatzdaten und variable Kosten Graz

Für den Standort Graz ist ebenfalls ein Profitcenter anzulegen. In Graz werden drei Arten von Sicherheitssystemen und zwei Ausführungen des selbst entwickelten Autoradios produziert.

Die Controlling-Abteilung erarbeitete folgende Zahlen für das Planungsjahr:

Sicherheitssysteme	SecuritySystem	SecuritySystem	SecuritySystem
	Small	Compact	Premium
Nettoumsatz	25.374.540,00 €	26.709.880,00 €	50.750.000,00 €
Umsatzsteuer	20 %	20 %	20 %
Var. Kosten (WES)	45 %	50 %	52 %
Rabatte	-	1 %	1,5 %
Skonto	-	2 %	1.015.000,00 €
Vertriebskosten	4 %	5 %	7 %
Umsatzprovision	1 %	1 %	2 %

Tabelle 15: Plandaten Graz

Autoradios	GreenRadio SF <sup>113</sup>	GreenRadio FF <sup>114</sup>
Absatzmenge Stk	109.249	275.727
Nettopreis	50,00 €	190,00 €
Umsatzsteuer	20 %	20 %
Var. Kosten (WES)	2.184.980,00 €	22.526.896,00 €
Rabatte	1,5 €/Stk	2 %
Skonto	2 %	1.238.474,00 €
So variable Kosten	4 %	2,0 €/Stk
Vertriebskosten	1,8 €/Stk	2,3 €/Stk

Tabelle 16: Plandaten Autoradios

	Sonst. betriebliche Erträge
Nettoumsatz	5.824.000,00 €

Tabelle 17: Sonst. betriebl. Erträge Graz

Die sonstigen betrieblichen Erträge werden über die Monate gleich.

Die Umsatzverteilung der Produkte basiert auf Erfahrungswerte aus den vorangegangenen Geschäftsjahren (in Stk):

### SecuritySystem Small:

<sup>113</sup> SF = Small Featured

<sup>114</sup> FF = Full Featured

<b>Jan.</b> 2 %	<b>Feb.</b> 5 %	<b>März</b> 7 %	<b>April</b> 6 %	<b>Mai</b> 10 %	<b>Juni</b> 11 %
<b>Juli</b> 10 %	<b>Aug.</b> 9 %	<b>Sept.</b> 8 %	<b>Okt.</b> 10 %	<b>Nov.</b> 10 %	<b>Dez.</b> 12 %

Tabelle 18: Produktionsmengen SecuritySystem Small

**SecuritySystem Compact:**

<b>Jan.</b> 3 %	<b>Feb.</b> 3 %	<b>März</b> 8 %	<b>April</b> 7 %	<b>Mai</b> 11 %	<b>Juni</b> 9 %
<b>Juli</b> 11 %	<b>Aug.</b> 10 %	<b>Sept.</b> 7 %	<b>Okt.</b> 9 %	<b>Nov.</b> 12 %	<b>Dez.</b> 10 %

Tabelle 19: Produktionsmengen SecuritySystem Compact

**SecuritySystem Premium:**

<b>Jan.</b> 4 %	<b>Feb.</b> 4 %	<b>März</b> 6 %	<b>April</b> 8 %	<b>Mai</b> 9 %	<b>Juni</b> 9 %
<b>Juli</b> 11 %	<b>Aug.</b> 9 %	<b>Sept.</b> 8 %	<b>Okt.</b> 9 %	<b>Nov.</b> 13 %	<b>Dez.</b> 10 %

Tabelle 20: Produktionsmengen SecuritySystem Premium

**GreenRadio SF:**

<b>Jan.</b> 4.370	<b>Feb.</b> 5.462	<b>März</b> 5.462	<b>April</b> 6.555	<b>Mai</b> 6.118	<b>Juni</b> 7.866
<b>Juli</b> 9.832	<b>Aug.</b> 9.395	<b>Sept.</b> 10.488	<b>Okt.</b> 10.925	<b>Nov.</b> 17.480	<b>Dez.</b> 15.295

Tabelle 21: Produktionsmengen GreenRadio SF

### GreenRadio FF:

<b>Jan.</b> 15.441	<b>Feb.</b> 18.749	<b>März</b> 22.058	<b>April</b> 23.161	<b>Mai</b> 26.470	<b>Juni</b> 24.264
<b>Juli</b> 19.852	<b>Aug.</b> 22.058	<b>Sept.</b> 19.852	<b>Okt.</b> 28.676	<b>Nov.</b> 28.676	<b>Dez.</b> 26.470

Tabelle 22: Produktionsmengen GreenRadio FF

### Fixe Kosten Graz

Auch in Graz können bestimmte Fixkosten direkt auf die Standorte aufgeteilt werden. Die Planfixkosten werden aus den Vorjahresdaten abgeleitet.

Personal	41.600.000,00 €
Miete	384.000,00 €
Sonstiger betrieblicher Aufwand	15.600.000,00 €

Tabelle 23: Fixkosten Graz

#### Personal:

Die Personalkosten werden auf die Monate gleichverteilt.

Einstellungen:

- Erfolgswirkung: ..... Aufwand Sonstiger
- Bilanzkonto: ..... So Verbindlichkeiten
- Detailkonto: ..... [Verbindlichkeiten Gehalt]

#### Miete:

In jedem Monat fallen gleich hohe Mietkosten an. Die Vorsteuer ist mit 20 % anzusetzen.

Einstellungen:

- Erfolgswirkung: ..... Aufwand Sonstiger
- Bilanzkonto: ..... Keine Zuordnung
- Detailkonto: ..... Keine Zuordnung

### Sonstiger betrieblicher Aufwand:

Die sonstigen betrieblichen Aufwendungen werden ebenfalls gleichverteilt, die Vorsteuer ist mit 20 % anzusetzen.

Einstellungen:

- Erfolgswirkung: ..... Aufwand Sonstiger
- Bilanzkonto: ..... So Verbindlichkeiten
- Detailkonto: ..... [So Verbindlichkeiten]

### Umsatzdaten und variable Kosten Salzburg

Die Zweigstelle in Salzburg wird als Servicestandort geführt. Einnahmen resultieren aus Serviceleistungen sowie aus sonstigen betrieblichen Erträgen.

Die Controlling-Abteilung erarbeitete folgende Zahlen für das Planungsjahr:

	Serviceleistungen	Sonst. betriebliche Erträge
Nettoumsatz	50.900,00 €	112.000,00 €
Vorsteuer	20 %	-
Var. Kosten (WES)	20 %	-

Tabelle 24: Plandaten Salzburg

Die Umsätze werden über die Monate gleichverteilt.

## Fixkosten Salzburg

In Salzburg können ebenfalls bestimmte Fixkosten direkt auf die Standorte aufgeteilt werden. Die Planfixkosten werden aus den Vorjahresdaten abgeleitet.

Personal	800.000,00 €
Miete	96.000,00 €
Sonstiger betrieblicher Aufwand	300.000,00 €

Tabelle 25: Fixkosten Salzburg

### Personal:

Die Personalkosten werden auf die Monate gleichverteilt.

Einstellungen:

- Erfolgswirkung: ..... Aufwand Sonstiger
- Bilanzkonto: ..... So Verbindlichkeiten
- Detailkonto: ..... [Verbindlichkeiten Gehalt]

### Miete:

In jedem Monat fallen gleich hohe Mietkosten an. Die Vorsteuer ist mit 20 % anzusetzen.

Einstellungen:

- Erfolgswirkung: ..... Aufwand Sonstiger
- Bilanzkonto: ..... Keine Zuordnung
- Detailkonto: ..... Keine Zuordnung

### Sonstiger betrieblicher Aufwand:

Die sonstigen betrieblichen Aufwendungen werden ebenfalls gleichverteilt, die Vorsteuer ist mit 20 % anzusetzen.

Einstellungen:

- Erfolgswirkung: ..... Aufwand Sonstiger
- Bilanzkonto: ..... So Verbindlichkeiten
- Detailkonto: ..... [So Verbindlichkeiten]

## Kostenstelle Verwaltung

Kosten, welche keinem Profitcenter direkt zugeordnet werden können, sind in der Kostenstelle Verwaltung zu erfassen.

	Verwaltung
Abschreibungen	16.323.000 €
Auflösung Abfertigungs-RST	125.000,00 €
Auflösung Pensions-RST	950.000,00 €

Tabelle 26: Kostenstelle Verwaltung

### Abschreibungen:

Der gesamte Abschreibungsbetrag wird im Dezember verbucht.

Einstellungen:

- Erfolgswirkung: ..... Neutraler Aufwand Abschreibung
- Bilanzkonto: ..... Anlagevermögen
- Detailkonto: ..... [Sachanlagen]

### Personal-RST:

Die beiden Posten „Auflösung Abfertigungs-RST“ und „Auflösung Pensions-RST“ sollen unter einem Aufwand-Ertrags-Element „Auflösung Personal-RST“ aufgegliedert werden.

Die Auflösung der Abfertigungs-RST erfolgt im Dezember. Die Pensions-RST werden im Juni mit 500.000,00 € und im Dezember mit 450.000,00 € aufgelöst.

Einstellungen Auflösung Personal-RST:

- Erfolgswirkung: ..... Neutraler Ertrag Sonstiger

Einstellungen Auflösungen Abfertigungs-RST:

- Erfolgswirkung: ..... Neutraler Ertrag Sonstiger
- Bilanzkonto: ..... Rückstellungen
- Detailkonto: ..... [Abfertigungsrückstellungen]

Einstellungen Auflösungen Pensions-RST:

- Erfolgswirkung: ..... Neutraler Ertrag Sonstiger
- Bilanzkonto: ..... Rückstellungen
- Detailkonto: ..... [Pensionsrückstellungen]

#### **Garantie-RST:**

Im Mai werden Garantierückstellungen in Höhe von 2.000.000,00 aufgelöst.

Einstellungen Auflösungen Garantie-RST:

- Erfolgswirkung: ..... Neutraler Ertrag Sonstiger
- Bilanzkonto: ..... Rückstellungen
- Detailkonto: ..... [So Rückstellungen]

### **A.3. Produktionsplanung**

#### **Lager**

Auf Unternehmensebene sind zwei Arten von Lager anzulegen:

- Teilelager
- Produktionslager

Auf dem Teilelager werden die zugekauften Elektronikbauteile gebucht.

Die produzierten Einheiten werden dem Produktionslager zugeschrieben.

## Produktionsplanung Wien

Folgende Zahlen werden für das Planjahr errechnet:

	Navi GlobalConnection	Navi EasyWay	Navi DWF
Produktionsmenge	176.549	307.570	132.514
Direkte Kosten	40,00 €/Einheit	38,00 €/Einheit	110,00 €/Einheit
Einkaufspreis Elektronikmodul	73,00 €	35,00 €	109,00 €
Vorsteuer	20 %	20 %	20 %
Gemeinkosten/ Elektronikmodul	10 %	14 %	10 %

Tabelle 27: Produktionsplanung Wien

Zur Vereinfachung wird angenommen, dass die jeweiligen Produktionsmengen über das Planjahr gleichverteilt werden. Pro produzierte Einheit wird ein Elektronikmodul verbaut, welches wir selbst über einen Lieferanten zukaufen.

Folgende Einstellungen gelten für alle drei Produkte:

Zuordnung Produktionslager: .....Produktionslager

Faktor 1 Gemeinkosten: .....Gemeinkosten %

Faktor 1 Vorsteuer: .....Vorsteuer %

Faktor 1 Bilanzkonto:.....Lager

Faktor 1 Detailkonto:.....Teilelager

## Produktionsplanung Graz

Für den Standort Graz stehen folgende Planzahlen zur Verfügung:

<b>Sicherheitssysteme</b>	SecuritySystem	SecuritySystem	SecuritySystem
	Small	Compact	Premium
Produktionsmenge	74.631	50.396	35.000
Direkte Kosten	103,00 €/Einheit	215,00 €/Einheit	664,00 €/Einheit
Einkaufspreis	45,00 €	45,00 €	80,00 €
Elektronikmodul			
Vorsteuer	20 %	20 %	20 %
Gemeinkosten/ Elektronikmodul	11,11 %	11 %	13 %

Tabelle 28: Produktionsplanung Sicherheitssysteme

<b>Autoradios</b>	GreenRadio SF	GreenRadio FF
Produktionsmenge	109.249	275.727
Direkte Kosten	10,00 €/Einheit	62,00 €/Einheit
Einkaufspreis	8,00 €	17,00 €
Elektronikmodul		
Vorsteuer	20 %	20 %
Gemeinkosten/ Elektronikmodul	25 %	18 %

Tabelle 29: Produktionsplanung Autoradios

Auch in Graz werden die Produktionsmengen über die Monate gleichverteilt. Pro produzierte Einheit wird ein Elektronikmodul verbaut, welches wir selbst über einen Lieferanten zukaufen.

Folgende Einstellungen gelten für alle fünf Produkte:

Zuordnung Produktionslager: .....Produktionslager

Faktor 1 Gemeinkosten: .....Gemeinkosten %

Faktor 1 Vorsteuer: .....Vorsteuer %

Faktor 1 Bilanzkonto:.....Lager

Faktor 1 Detailkonto:.....Teilelager

## **A.4. Finanzplanung und Planbilanz**

### **Konten anlegen**

Nachstehende Konten müssen für das Planjahr angelegt werden.

Die Anfangsbestände ergeben sich durch die Endbestände des letzten Jahres.

### **Anlagevermögen**

#### **Sachanlagen**

Der Anfangsbestand der Sachanlagen beträgt 425.000.000,00.

Einstellungen:

- Anlagevermögen Verb Hauptbilanz: .....Verb LuL
- Anlagevermögen Verb Detailkonto: .....[Verbindlichkeiten Inland]

#### **Finanzanlagen**

Der Anfangsbestand der Finanzanlagen beläuft sich auf 437.000,00.

Einstellungen:

- Anlagevermögen Verb Hauptbilanz: .....Verb LuL
- Anlagevermögen Verb Detailkonto: .....[Verbindlichkeiten Inland]

## **Verpackungsmaschine und Spritzgussmaschine**

Im Laufe des folgenden Jahres werden zwei Anlagen angeschafft:

- Verpackungsmaschine
- Spritzgussmaschine

## **Verbindlichkeiten LuL**

### **Verbindlichkeiten Verpackungsmaschine und Spritzgussmaschine**

Um die beiden Investitionen abbilden zu können werden folgende Verbindlichkeiten-Konten benötigt:

- Verbindlichkeiten Verpackungsmaschine (Absolutwert)
- Verbindlichkeiten Spritzgussmaschine (Absolutwert)

### **Verbindlichkeiten Inland**

Das Konto Verbindlichkeiten Inland (Z-Ziel 30, AB 320.000,00) wird angelegt um die allgemeinen Verbindlichkeiten abzubilden.

Der Anfangsbestand wird im Jänner abgebaut.

Einstellungen:

- Verbindlichkeiten LuL: .....Zahlungsziel
- Anfangsbestand: .....320.000,00

## **Sonstige Verbindlichkeiten**

### **Verbindlichkeiten Personal**

Die beiden Elemente

- Verbindlichkeiten Gehalt
- Verbindlichkeiten Provisionen (Z-Ziel 30, AB 20.000,00)

werden unter dem Element „Verbindlichkeiten Personal“ aufgliedert.

Einstellungen Verbindlichkeiten Personal:

So Verbindlichkeiten: ..... Umschlag/Stichtag

Einstellungen Verbindlichkeiten Gehalt:

So Verbindlichkeiten: ..... Absolutwert

Einstellungen Verbindlichkeiten Provisionen:

So Verbindlichkeiten: ..... Zahlungsziel

### **So Verbindlichkeiten**

Das Konto So Verbindlichkeiten wird mit 7 Tagen Zahlungsziel eingerichtet.

## **Rückstellungen**

### **Personalarückstellungen**

Alle Rückstellungen das Personal betreffend werden unter dem Element Personalarückstellungen zusammengefasst.

Darunter sind folgende Rückstellungskonten anzulegen:

- Abfertigungsrückstellungen
- Pensionsrückstellungen
- RST für Jubiläumsgelder

Zu Beginn des Jahres betragen die Anfangsbestände der

- Abfertigungsrückstellungen 800.000,00 €
- Pensionsrückstellungen 10.500.000,00 €
- RST für Jubiläumsgelder 600.000,00 €.

### **Sonstige Rückstellungen**

Auf dem Konto Sonstige Rückstellungen werden alle übrigen Rückstellungen zusammengefasst. Der Anfangsbestand beträgt 15.000.000,00 €.

## **Aktive Rechnungsabgrenzungen**

Auf dem Element ARAP<sup>115</sup> sind die Anfangsbestände der Aktiven Rechnungsabgrenzung sowie Veränderungen der Konten zu erfassen.

Der Anfangsbestand der ARAP beläuft sich auf 400.000,00 €. Im Jänner werden 250.000,00 € und im Februar die restlichen 150.000,00 € abgebaut.

Im Dezember werden neue aktive Rechnungsabgrenzungen in der Höhe von 500.000,00 € erfasst.

## **Passive Rechnungsabgrenzungen**

Auf dem Element PRAP<sup>116</sup> sind die Anfangsbestände der Passiven Rechnungsabgrenzungen sowie Veränderungen der Konten zu erfassen.

Der Anfangsbestand der PRAP beträgt 320.000,00 €. 180.000,00 € werden im Jänner und 140.000,00 € im Februar aufgelöst. Im Dezember sind passive Rechnungsabgrenzungen in der Höhe von 130.000,00 € zu erfassen.

## **Eigenkapital**

### **Gewinnrücklage**

Die Gewinnrücklage aus dem Vorjahr beträgt 250.000,00 €. Im November wird eine Gewinnrücklage in der Höhe von 200.000,00 € gebildet.

---

<sup>115</sup> ARAP = Aktive Rechnungsabgrenzungsposten

<sup>116</sup> PRAP = Passive Rechnungsabgrenzungsposten

### **Stammkapital**

Das Eigenkapital beträgt 11.500.000,00 €. Im März beschließen die Gesellschafter eine Einlage in der Höhe von 1.500.000,00 € zu tätigen.

### **Sonstiges Umlaufvermögen**

Auf dem Element So Umlaufvermögen werden die sonstigen Umlaufvermögensgegenstände zusammengefasst. Zu Beginn des Jahres beträgt die Höhe des sonstigen Umlaufvermögens 200.000,00 €. Im Februar und Mai findet ein Abbau von 60.000,00 € bzw. 38.000,00 statt. Im Oktober erhöht sich das sonstige Umlaufvermögen um 52.000,00.

### **Forderungen LuL**

Die Controlling-Abteilung analysierte die Forderungsentwicklung der letzten Jahre und kam zu folgendem Schluss:

Nur 10 % der ausstehenden Forderungen werden innerhalb von 7 Tagen bezahlt. 30 % werden in 14 Tagen beglichen. Über die Hälfte (60 %) zahlt innerhalb von 30 Tagen.

### **Kalkulatorische Kosten**

Unter der Kostenstelle Verwaltung ist ein Strukturelement vom Typ kalkulatorische Kosten mit der Bezeichnung kalkulatorische Afa anzulegen. Auf diesem Konto werden alle kalkulatorischen Abschreibungen erfasst.

### **Darlehen**

Auf Unternehmensebene ist ein Strukturelement vom Typ Darlehen (Investitionsdarlehen) anzulegen.

## A.5. Investitionen

Für das kommende Jahr sind zwei große Investitionen zu planen.

### Wien – Spritzgussmaschine

In Wien werden im Februar für die Produktion der Navigationsgeräte neue Spritzgussmaschinen angeschafft.

Anschaffungswert: 6.170.700,00 €  
Nutzungsdauer: 7 Jahre

Für die Errechnung der kalkulatorischen Abschreibung wird eine kalkulatorische Nutzungsdauer von 10 Jahren angenommen.

Die Maschine wird erst im März in Betrieb genommen.

Für die Erfassung der buchhalterischen Abschreibung ist ein Strukturelement vom Typ Aufwand/Ertrag (Afa Spritzgussmaschine) in der Niederlassung Wien anzulegen.

Einstellungen:

- Erfolgswirkung: .....Aufwand Abschreibung
- Bilanzkonto: .....Anlagevermögen
- Detailkonto: .....Spritzgussmaschine

Um diesen Investitionsvorgang erfassen zu können ist ein Element vom Typ Investition mit dem Namen Investition Spritzgussmaschine unter dem Profitcenter Wien anzulegen.

- ..... Einstellungen:

- Afa Einstellungen pag: .....Degressive/Lineare Afa Quote
- Afa Einstellungen kalk: .....Lineare Afa
- Afa pag Zuordnung: .....Afa Spritzgussmaschinen
- Afa kalk Zuordnung: .....kalkulatorische Afa
- Investitionszuordnung: .....Spritzgussmaschinen

Für die Finanzierung der Spritzgussmaschinen wird ein Kredit über die volle Investitionssumme aufgenommen. Hierfür ist zuerst ein Aufwand/Ertrags-Konto (Aufwand Kreditzinsen), um die Zinsen des Kredits abbilden zu können, anzulegen.

Einstellungen:

- Erfolgswirkung: .....Aufwand Zinsen
- Bilanzkonto: .....So Verbindlichkeiten
- Detailkonto: .....So Verbindlichkeiten

Der aufgenommene Kredit wird mittels eines Kredit-Elements (Kredit Spritzgussmaschine) abgebildet.

Einstellungen:

- Zinsfälligkeit: ..... Quartale
- Tilgungsfrequenz: ..... Monate
- Kreditart: ..... Annuität (ZnT, N)<sup>117</sup>
- Kredit Bilanzkonto AB: ..... Darlehen
- Kredit Detailkonto AB: ..... Investitionsdarlehen
- Kredit Bilanzkonto Veränderungen: ..... Darlehen
- Kredit Detailkonto Veränderungen: ..... Investitionsdarlehen
- Zinsen Zuordnung: ..... Aufwand Kreditzinsen

## **Graz – Verpackungsmaschine**

Am Standort Graz wird im April eine neue Verpackungsmaschine angeschafft.

Anschaffungswert: ..... 4.113.800,00 €

Nutzungsdauer: ..... 7 Jahre

Für die Errechnung der kalkulatorischen Abschreibung wird eine kalkulatorische Nutzungsdauer von 10 Jahren angenommen.

Die Maschine wird erst im Mai in Betrieb genommen.

---

<sup>117</sup> ZnT = Zinsen nach Tilgung, N = nachschüssig

Für die Erfassung der buchhalterischen Abschreibung ist ein Strukturelement vom Typ Aufwand/Ertrag (Afa Verpackungsmaschine) in der Niederlassung Graz anzulegen.

Einstellungen:

- Erfolgswirkung: .....Aufwand Abschreibung
- Bilanzkonto: .....Anlagevermögen
- Detailkonto: .....Verpackungsmaschine

Um diesen Investitionsvorgang erfassen zu können ist ein Element vom Typ Investition mit dem Namen Investition Verpackungsmaschine unter dem Profitcenter Graz anzulegen.

Einstellungen:

- Afa Einstellungen pag: .....Degressive/Lineare Afa Quote
- Afa Einstellungen kalk: .....Lineare Afa
- Afa pag Zuordnung: .....Afa Verpackungsmaschine
- Afa kalk Zuordnung: .....kalkulatorische Afa
- Investitionszuordnung: .....Verpackungsmaschine

Für die Finanzierung der Verpackungsmaschine wird ein Kredit über die volle Investitionssumme aufgenommen. Hierfür ist zuerst ein Aufwand/Ertrags-Konto (Aufwand Kreditzinsen), um die Zinsen des Kredits abbilden zu können, anzulegen.

Einstellungen:

- Erfolgswirkung: .....Aufwand Zinsen
- Bilanzkonto: .....So Verbindlichkeiten
- Detailkonto: .....So Verbindlichkeiten

Der aufgenommene Kredit wird mittels eines Kredit-Elements (Kredit Verpackungsmaschine) abgebildet.

Einstellungen:

- Zinsfälligkeit: .....Monate
- Tilgungsfrequenz: .....Monate
- Kreditart: .....Annuität (ZvT, V)<sup>118</sup>
- Kredit Bilanzkonto AB: .....Darlehen

---

<sup>118</sup> ZvT = Zinsen vor Tilgung, V = vorschüssig

- Kredit Detailkonto AB: ..... Investitionsdarlehen
- Kredit Bilanzkonto Veränderungen: ..... Darlehen
- Kredit Detailkonto Veränderungen: ..... Investitionsdarlehen
- Zinsen Zuordnung: ..... Aufwand Kreditzinsen

## **A.6. Steuern**

Im vorliegenden Szenario wird nur die Körperschaftssteuer berücksichtigt.

### **Körperschaftssteuer**

Der Körperschaftssteuersatz beträgt 25 %.

Steuervorauszahlungen sind keine zu berücksichtigen.