



HOCHSCHULE KONSTANZ TECHNIK, WIRTSCHAFT UND GESTALTUNG
UNIVERSITY OF APPLIED SCIENCES

ANLO-Projekt SS13 WIN6 Maschinenbelegungsplanung HTWG Konstanz

Prof. Dr. **Michael Grütz**

Dokumentation der Projektarbeit über Migration des Tools
Maschinenbelegungsplanung

Burcu Albostan, Elena Oberländer

WIN 6

285 145, 284 862

20.06.2013

Inhaltsverzeichnis

1.	Einleitung.....	2
2.	Beschreibung IST-Zustand	3
3.	Zielvorgaben	5
4.	Umsetzung.....	6
4.1.	Solverpfad und Arbeitsverzeichnis	6
4.2.	Erweiterung der Ausgabeoptionen	7
4.3.	Graphische Darstellung	8
5.	Neuer IST-Zustand	9
6.	Budgetauswertung	13
7.	Fazit	14

1. Einleitung

Zu Beginn kann gesagt werden, dass die Methode Maschinenbelegungsplanung vor Projektstart des Zweierteams (Elena Oberländer und Burcu Albostan) unter Windows 7 teilweise funktionsfähig war und diverse Fehler aufzeigte.

Das Ziel des Projektes war es, das C#-Programm „Maschinenbelegungsplanung“ unter Windows 7 64-Bit vollfunktional lauffähig zu machen.

Am Ende des Projektes wurden die Fehler behoben. Die Funktionalität Berechnung für eine oder mehrere Maschinen und die zukünftige Nutzung auf diesem Betriebssystem wurden ermöglicht. Studenten können weitere Verbesserungen am Programm vornehmen.

2. Beschreibung IST-Zustand

Im Folgenden wird kurz der IST-Zustand des Tools Maschinenbelegungsplanung beschrieben, wie er vor der Überarbeitung war.

Die Methode lässt sich unter Windows 7 starten. Der Nutzer kann über eine Menüleiste auswählen ob er die Daten importieren möchte. Es ist auch möglich, die gewünschten Daten manuell einzutragen. Die eingegebenen Werte sollten dann über die Funktion Berechnen berechnet und das Ergebnis auf dem Bildschirm ausgegeben werden. Jedoch gibt die Methode verschiedene Fehlermeldungen aus. Das Berechnen für eine oder mehrere Maschinen kann nicht durchgeführt werden. Bei der Ausgabeoption ist die Einstellung bereits Numerisch ausgewählt, eine andere Option ist nicht auswählbar.

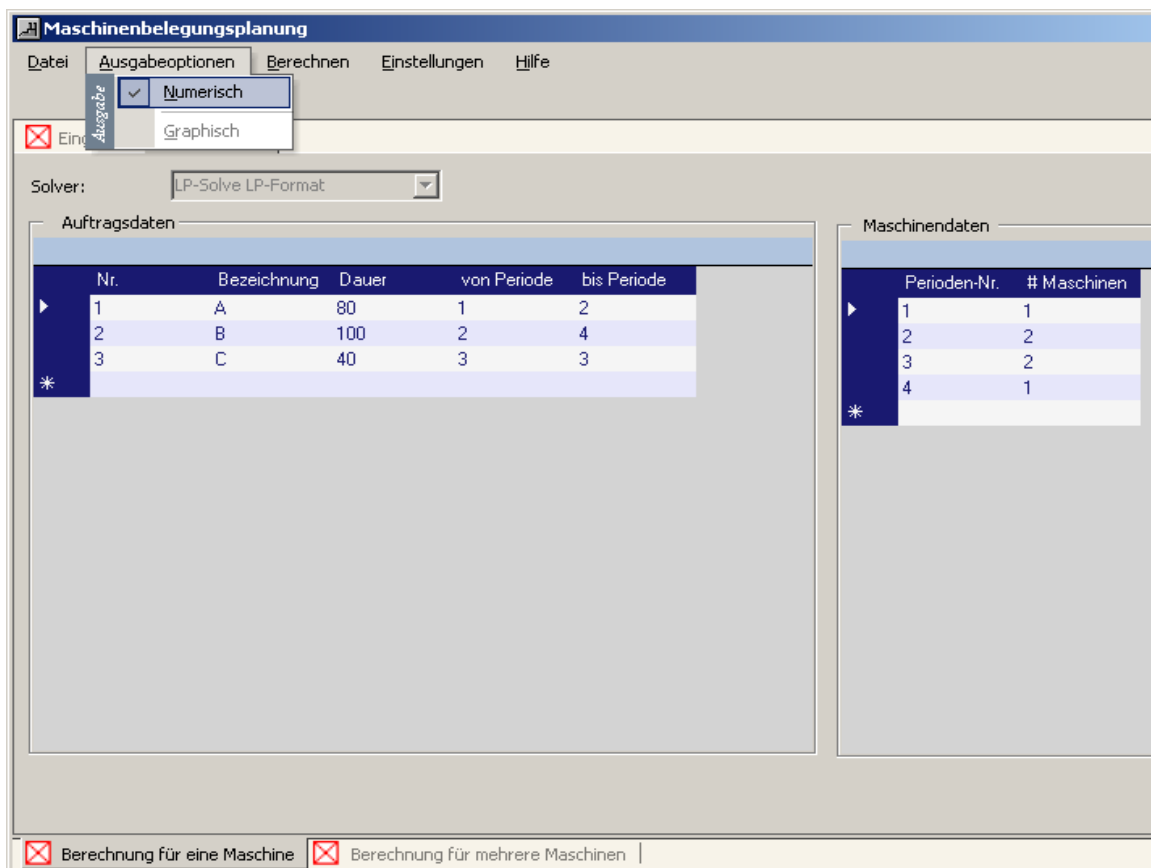


Abbildung 1: Ausgabeoptionen

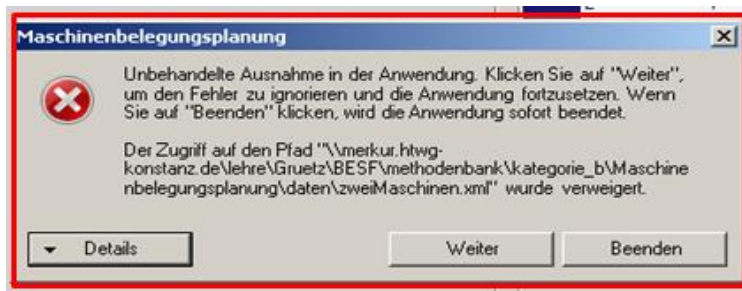


Abbildung 2: Fehlermeldung 1

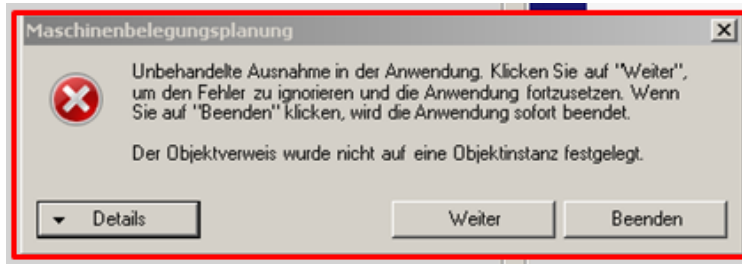


Abbildung 3: Fehlermeldung 2

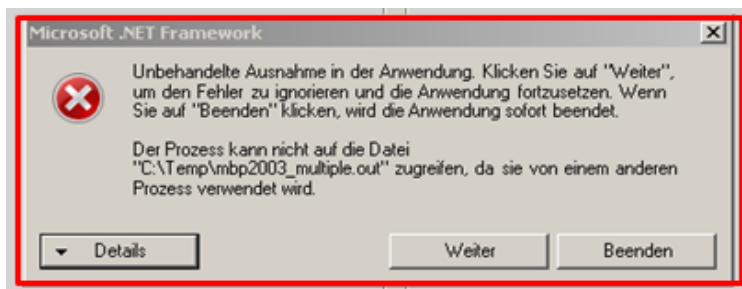


Abbildung 4: Fehlermeldung 3

3. Zielvorgaben

Als Ziel wurde sich gesetzt, diese Fehler zu beheben und die Berechnung für eine oder mehrere Maschinen zu ermöglichen.

Ebenso musste eine Fehleranalyse durchgeführt und eventuell das Programm in C# weiterentwickelt werden. Um eine Qualitätssicherung durchzuführen, wurde das Tool auf Lauffähigkeit unter Windows 7 (64 Bit) getestet.

Zusätzlich sollte eine Ausgabeoption hinzugefügt werden, um die Ergebnisse zu visualisieren.

4. Umsetzung

4.1.Solverpfad und Arbeitsverzeichnis

Wie bereits erwähnt kamen bei der Berechnung verschiedene Fehlermeldungen auf. Nach längerer Fehleranalyse des Quellcodes sind wir auf das Problem gestoßen, dass beim Ausführen der Berechnungen auf die Solverdateien nicht zugegriffen werden kann. Für die Berechnungen ist der LP-Solver notwendig.

Die Pfade im Programmcode des alten Tools waren hart reincodiert. Bei der Änderung von Pfaden müssten diese im Quellcode geändert werden, um Zugriff auf Solverdateien zu ermöglichen. Um dieses Problem umzugehen haben wir eine Konfigurationsdatei erstellt. Über diese Datei können die Pfade für die Solver und das Arbeitsverzeichnis geändert werden. Dabei muss der Code nicht neu kompiliert werden. Der Solverpfad und das Arbeitsverzeichnis können entweder in Visual Studio

	Name	Typ	Bereich	Wert
	SolverPfad	string	Benutzer	C:\Maschinenbelegungsplanung\lp_solve\EXEC
	Arbeitsverzeich...	string	Benutzer	C:\temp
*				

Abbildung 5: Konfigurationsdatei in Visual Studio

oder in einem Editor z.B. Notepad festgelegt werden.

```

1  <?xml version="1.0" encoding="utf-8" ?>
2  <configuration>
3    <configSections>
4      <sectionGroup name="userSettings" type="System.Configuration.UserSettingsGroup, System, Version=2.0.0.0, Culture=neutral, PublicKeyToken
5        <section name="Maschinenbelegungsplanung.Einstellungen" type="System.Configuration.ClientSettingsSection, System, Version=2.0.0.0, C
6      </sectionGroup>
7    </configSections>
8    <userSettings>
9      <Maschinenbelegungsplanung.Einstellungen>
10       <setting name="SolverPfad" serializeAs="String">
11         <value>C:\Maschinenbelegungsplanung\lp_solve\EXEC</value>
12       </setting>
13       <setting name="Arbeitsverzeichnis" serializeAs="String">
14         <value>C:\temp</value>
15       </setting>
16     </Maschinenbelegungsplanung.Einstellungen>
17   </userSettings>
18 </configuration>

```

Abbildung 6: Konfigurationsdatei im Editor

Wie man in der oberen Abbildung sehen kann, enthält die Konfigurationsdatei den elementarsten XML-Code.

In der Klasse MainForm werden die Werte ausgelesen.

```
private string fileName = null;
//Solverpfad wird nun über Einstellungen.xml Datei eingelesen
public static string solverPath = Einstellungen.Default.SolverPfad;
//-----
//Arbeitsverzeichnis wird nun über Einstellungen.xml Datei eingelesen
public static string workingDir = Einstellungen.Default.Arbeitsverzeichnis;
//-----
private System.Windows.Forms.ImageList imageList1;
private PrintDocument printDocument = null;
```

Abbildung 7: Einlesen des Solverpfades und des Arbeitsverzeichnisses in MainForm

Die Datei Einstellungen ist im Programmordner "exec" zu finden.

book.ico	11.01.1996 23:00	Symbol	2 KB
factory.ico	28.11.2002 20:55	Symbol	2 KB
help.ico	11.01.1996 23:00	Symbol	2 KB
input.ico	11.01.1996 23:00	Symbol	2 KB
lighton.ico	11.01.1996 23:00	Symbol	2 KB
MagicLibrary.DLL	03.11.2002 12:51	Anwendungserweit...	388 KB
Maschinenbelegungsplanung.exe	26.06.2013 12:51	Anwendung	92 KB
Maschinenbelegungsplanung.exe.config	26.06.2013 12:59	XML Configuration File	1 KB
Maschinenbelegungsplanung.pdb	26.06.2013 12:51	Program Debug Dat...	176 KB
Maschinenbelegungsplanung.vshost.exe	26.06.2013 13:00	Anwendung	12 KB
Maschinenbelegungsplanung.vshost.exe.config	26.06.2013 12:59	XML Configuration File	1 KB
Maschinenbelegungsplanung.vshost.exe.ma...	17.03.2010 22:39	MANIFEST-Datei	1 KB
question.ico	11.01.1996 23:00	Symbol	2 KB
result.ico	11.01.1996 23:00	Symbol	2 KB
table.ico	17.09.1996 00:00	Symbol	1 KB
ZedGraph.dll	12.12.2008 15:01	Anwendungserweit...	300 KB

Abbildung 8: Programmordner "exec"

4.2. Erweiterung der Ausgabeoptionen

In der alten Version wurde die grafische Darstellung der Ergebnisse nicht implementiert. Es stand nur die numerische Ausgabeoption zur Verfügung. Wir haben uns dazu entschlossen, eine grafische Ausgabeoption für das Visualisieren der Ergebnisse erzeugen zu lassen.

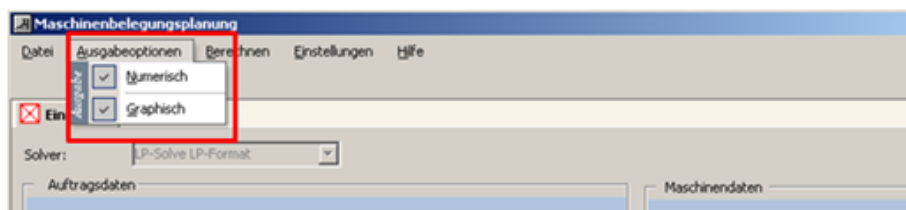


Abbildung 9: Graphische Ausgabeoptionen verfügbar


```
protected void CreateViewMenu(MenuCommand mc)
{
    // Create menu commands
    MenuCommand viewNumeric = new MenuCommand("&Numerisch", new EventHandler( onViewNumeric.Checked = true;

    //Änderung im Rahmen der ANLO Veranstaltung. Graphische Ausgabeoption verfügbar
    MenuCommand viewSplitterOne = new MenuCommand("-");
    MenuCommand viewGraphical = new MenuCommand("&Graphisch");
    viewGraphical.Enabled = true;
    viewGraphical.Checked = true;
```

Abbildung 10: Programmcode für die graphische Ausgabeoption

4.3. Graphische Darstellung

Wie oben schon erwähnt wurde, haben wir die Methode für die graphische Darstellung erweitert. Dafür haben wir die Klasse Graphics implementiert.

Nach der Berechnung werden die Ergebnisse in einem Array - `double[,] sinlgeSolverResults` - in der Klasse ResultForm gespeichert. Hier wird auch die Methode `erstelleGraphic2` mit den Übergabeparameter `sinlgeSolverResults`:

```
gr.erstelleGraphic2(sinlgeSolverResults);
gr.Show();
```

Abbildung 11: Aufruf "erstelleGraphic2" für eine Maschine

Bei der Berechnung für zwei Maschinen werden die Ergebnisse im Array - `double[, ,]` `multipleSolverResults` - gespeichert. Aufruf für die Methode:

```
gr.erstelleGraphic(multipleSolverResults);
gr.Show();
```

Abbildung 12: Aufruf Methode "erstelleGraphic" für zwei Maschinen

Der gesamte Code ist unter dem Verzeichnis `source/Forms` zu finden.

Die graphische Ausgabe für eine Maschine stellt die Auslastung einer Maschine, die auf Perioden verteilt ist.

Bei der Berechnung für zwei Maschinen wird die Auslastung der ersten bzw. der zweiten Maschine pro Auftrag dargestellt.

5. Neuer IST-Zustand

Nach allen vorgenommenen Änderungen ist der neue IST-Zustand des Programms Maschinenbelegungsplanung gemäß des Commitments. Die Berechnung für eine oder mehrere Maschinen ist funktionsfähig, das Visualisieren der Ergebnisse über die graphische Ausgabeoption wurde hinzugefügt, Pfade können nun über eine XML Datei eingelesen und verwaltet werden, selbst wenn keine Microsoft Visual Studio Kenntnisse vorhanden sind.

Im Rahmen der Änderungen wurde ebenfalls die Oberfläche entsprechend angepasst. Die neue Startoberfläche sieht damit wie folgt aus.

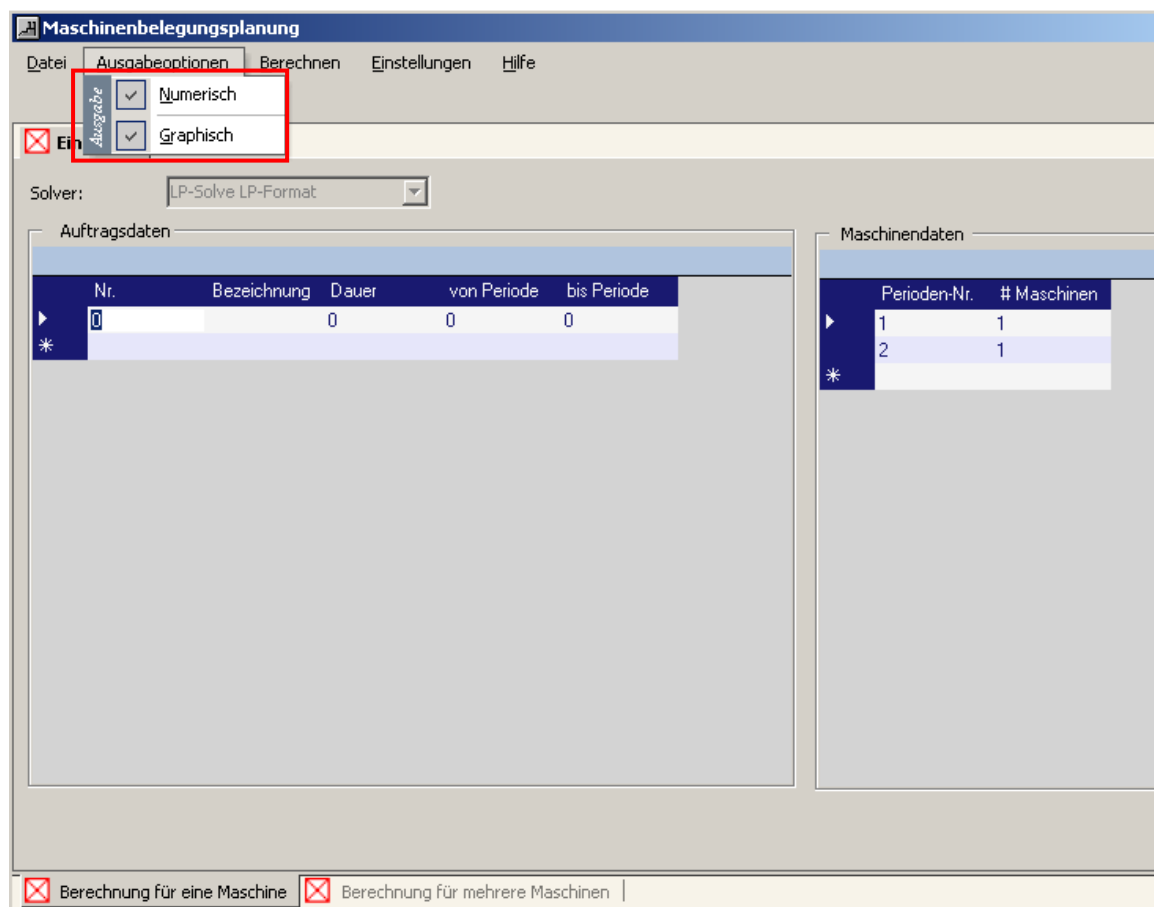


Abbildung 13: Oberfläche: erweiterte Ausgabeoptionen

Außer der graphischen Ausgabeoption, wurde keine weitere Änderung an der Oberfläche vorgenommen.

Alle Funktionen wie Laden und Speichern sind vollfunktionsfähig. Alle Programmänderungen sind schriftlich in diesem Dokument festgehalten oder im Quellcode kenntlich gemacht durch z.B. Kommentare.

Das Programm Maschinenbelegungsplanung ist somit vollständig unter Windows 7 lauffähig.

Maschinenbelegungsplanung

Datei Ausgabeoptionen Berechnen Einstellungen Hilfe

☒ Eingabe 1 ☒ **Eingabe 2**

Solver: LP-Solve LP-Format

Auftragsdaten

Nr.	Bezeichnung	Dauer	von Periode	bis Periode
1	A	80	1	2
2	B	100	2	4
3	C	40	3	3

Maschinendaten

Perioden-Nr.	# Maschinen
1	1
2	2
3	2
4	1

☒ Berechnung für eine Maschine ☒ Berechnung für mehrere Maschinen

Abbildung 14: Berechnung für eine Maschine - Eingabewerte

Maschinenbelegungsplanung

Datei Ausgabeoptionen Berechnen Einstellungen Hilfe

☒ Eingabe 1 ☒ Eingabe 2 ☒ **Ergebnis**

Durchschnittliche Maschinenauslastung pro Periode: 36.6666667

	Auftrag 1	Auftrag 2	Auftrag 3
Periode 1	36,667	0	0
Periode 2	43,333	30	0
Periode 3	0	33,333	40
Periode 4	0	36,667	0

☒ Berechnung für eine Maschine ☒ Berechnung für mehrere Maschinen

Abbildung 15: Berechnung für eine Maschine - Ergebnis

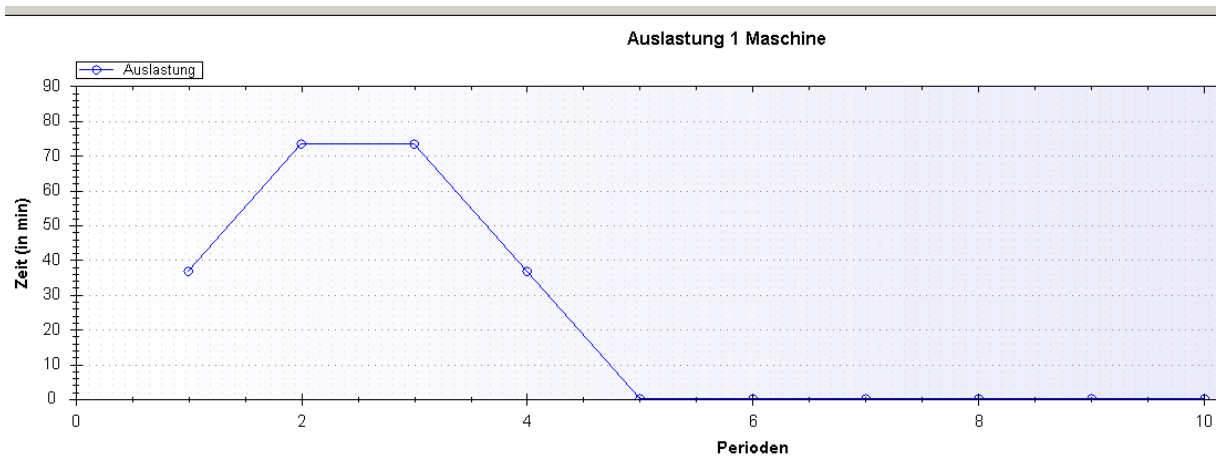


Abbildung 16: Berechnung für eine Maschine - Graphische Ausgabe

Maschinenbelegungsplanung

☒ Eingabe 1
 ☒ Eingabe 2

Solver:

Auftragsdaten

Nr.	Bezeichnung	Dauer	Typ 1	Typ 2	von Periode
1	A	80	1	0	1
2	B	100	1	0	2
3	C	50	1	1	3

Maschinendaten

Perioden-Nr.	# Maschinen	# Maschinen
1	1	0
2	2	0
3	1	1
4	1	1

☒ Berechnung für eine Maschine
 ☒ Berechnung für mehrere Maschinen

Abbildung 17: Berechnung für mehrere Maschinen - Eingabewerte

Maschinenbelegungsplanung

☒ Datei
 ☒ Ausgabeoptionen
 ☒ Berechnen
 ☒ Einstellungen
 ☒ Hilfe

☒ Eingabe 1
 ☒ Eingabe 2
 ☒ **Ergebnis**

Durchschnittliche Maschinenauslastung pro Periode: 36

	Auftrag 1	Auftrag 2	Auftrag 3
Periode 1 Typ 36	0	0	0
Periode 1 Typ 0	0	0	0
Periode 2 Typ 44	0	0	0
Periode 2 Typ 0	0	0	0
Periode 3 Typ 0	0	0	0
Periode 3 Typ 0	0	0	0
Periode 4 Typ 0	0	0	0
Periode 4 Typ 0	0	0	0
Periode 1 Typ 0	0	0	0
Periode 1 Typ 0	0	0	0
Periode 2 Typ 0	28	0	0
Periode 2 Typ 0	0	0	0
Periode 3 Typ 0	36	0	0
Periode 3 Typ 0	0	0	0
Periode 4 Typ 0	36	0	0
Periode 4 Typ 0	0	0	0
Periode 1 Typ 0	0	0	0

☒ Berechnung für eine Maschine
 ☒ Berechnung für mehrere Maschinen

Abbildung 18: Berechnung für mehrere Maschinen - Ergebnis

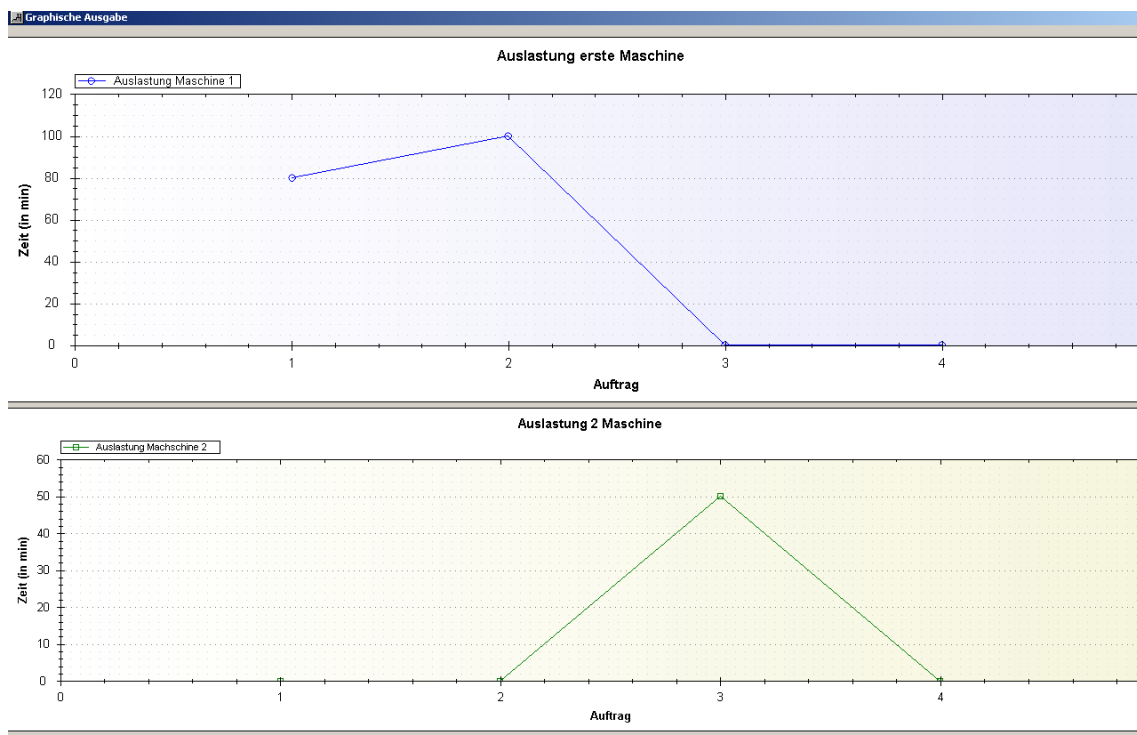


Abbildung 19: Berechnung für mehrere Maschinen - Graphische Ausgabe

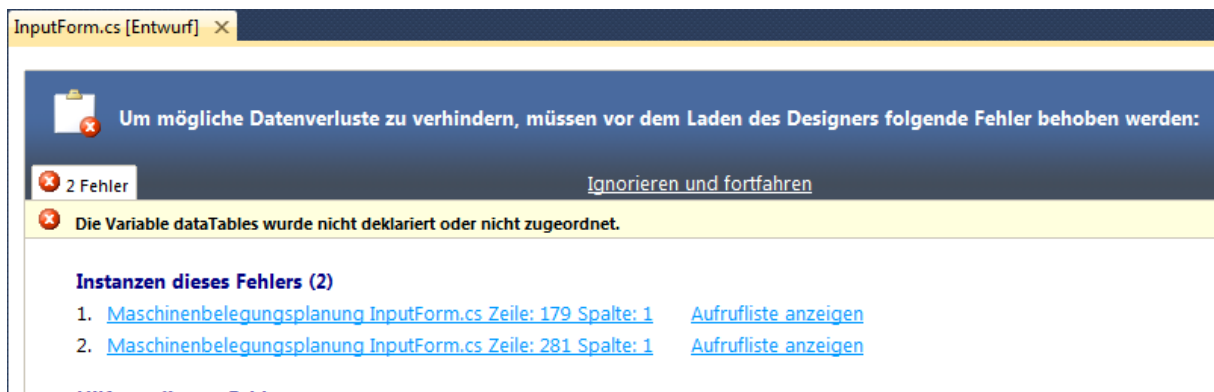
6. Budgetauswertung

Die Budgetauswertung in diesem Projekt sieht so aus, dass 60 Stunden pro Person zur Verfügung stehen. In Real wurden ungefähr 40 Stunden für die Programmierung aufgewandt (pro Person). Anschließend folgten die Fehleranalyse, das Testing und das Schreiben der Dokumentation, welche wiederum 10 Stunden in Anspruch nahmen. Wie zu sehen ist, konnten und mussten die zur Verfügung stehenden 60 Stunden nicht ganz ausgeschöpft werden. Jedoch ist zu beachten, dass immer mit unerwarteten Komplikationen zu rechnen ist, welche weitaus mehr Zeit in Anspruch nehmen können, was in diesem Projekt glücklicherweise nicht der Fall war. Ein durchdachter Start des Projektes konnte bereits viele Risiken und Fehlerquellen ausschließen.

7. Fazit

Als Fazit lässt sich abschließend sagen, dass unser Projekt erfolgreich abgeschlossen und an das Teamprojekt übergeben werden konnte. Die Fehleranalyse des Tools hat sich jedoch aufgrund des Programmumfangs und der teilweise fehlerhaften Dokumentation als sehr schwierig herausgestellt. Wir brauchten insgesamt deutlich länger für die Fehleranalyse als erwartet und mussten somit die Programmierarbeiten in weniger Zeit erledigen als geplant.

Bei der Umsetzung haben wir die Fehlermeldung bekommen:



Nach der Recherche haben wir herausgefunden, dass die Designerdatei eine oder mehrere Fehler enthält. Diese führen dazu, dass Formulare nicht mehr in der Designansicht angezeigt werden können. Oft kann das Programm weiter ausgeführt werden, wie in unserem Fall, aber eine Verbesserung ist hierbei möglich. Ebenso kann eine Verbesserung der graphischen Ausgabe vorgenommen werden.

Trotz aller Probleme war es ein insgesamt erfolgreiches Projekt, welches im Rahmen des vorhanden Budgets abgeschlossen werden konnte.