## Realisierung & Test

### Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 16.04.2012 | 1.0 | Erste Version des Dokuments | DT |
| 22.04.2012 | 1.1 | Usability Tests | CH |
| 23.04.2012 | 1.2 | Systemtests | DT |
| 24.04.2012 | 1.3 | Review | DT |
| 27.04.2012 | 1.4 | Systemtests | DT |
| 07.05.2012 | 1.5 | Systemtests | DT |
| 11.05.2012 | 1.6 | Systemtests | DT |
| 12.05.2012 | 1.7 | Einfügen Testdokumentation aus Domain Analyse | DT |
| 14.05.2012 | 1.8 | Systemtests | DT |
| 18.05.2012 | 1.9 | Usability Test: Reaktion auf Demomodus | CH |
| 18.05.2012 | 1.10 | Review | DT |
| 18.05.2012 | 1.11 | Extension Framework | LE |
| 19.05.2012 | 1.12 | Ergänzung Wizard of Oz Test mit Bild | DT |
| 22.05.2012 | 1.13 | Review Extension Framework | DT |
| 22.05.2012 | 1.14 | Systemtests | DT |
| 22.05.2012 | 1.15 | Usability Test: Grafisches Design | DT |
| 23.05.2012 | 1.16 | Coding Standards & Dokumentation Quellcode | CH |
| 23.05.2012 | 1.17 | Review Usability Test: Grafisches Design | CH |

### Realisierung

#### Framework

##### Grundlagen

Die erste Frage, die sich bei einem Framework stellt, ist, wie eine Extension in das Framework geladen wird. Microsoft bietet für diesen Zweck das Managed Extensibility Framework (MEF)[[1]](#footnote-1) an. Technische Details dazu können in der MEF-Dokumentation[[2]](#footnote-2) nachgelesen werden.

Die wichtigste Funktionalität von MEF, die für die HSR Video Wall gebraucht wird, ist die folgende:

* Das Schlüsselwort Export zeichnet eine Klasse (Einstiegspunkt), die ein von einem Framework (HSR Video Wall Applikation) definierten Interface (IApp) implementiert, für den Export.
* Das Framework (HSR Video Wall Applikation) importiert alle Klassen, die einen bestimmten Pfad haben (Ordner Extensions) und das Interface IApp implementieren.



Abbildung - Poster Applikation (Extension) wird über [Export(typeof(IApp))] als IApp exportiert

Die obenstehende Abbildung zeigt die Klasse PosterApp, welche das Interface IApp implementiert. Der Ausdruck [Export(typeof(IApp))] markiert die Klasse für den Export.



Abbildung - Die Video Wall Applikation (Framework) importiert über [ImportMany(AllowRecomposition = false)] alle Klassen, die das Interface IApp implementieren

Der Ausdruck [ImportMany(AllowRecomposition = false)] im Framework (Video Wall Applikation) importiert alle Klassen, welche das Interface IApp implementieren.

##### Probleme

Beim Entwickeln eines Frameworks ist oftmals nicht vorhersehbar, wie dieses in der Zukunft aussehen wird, da sich die Anforderungen an das Framework stetig ändern. Würde nur ein einziges Interface (IApp), über das die Services des Frameworks angesprochen werden können, zur Verfügung gestellt, so müsste sich dieses ständig ändern. Folglich müssten die Extensions, zum Beispiel die PosterApp (siehe Unterkapitel I.1.2.1.1 Grundlagen), nach jeder Änderung am Interface (IApp) neu kompiliert werden. Mit nur einem Interface ist es also schwierig, den Extensions neue Funktionalität zur Verfügung zu stellen.   
Ein weiteres Problem eines einzigen Interfaces ist, dass dieses beliebig gross werden kann und dadurch die Kopplung steigt und die Kohäsion sinkt, was sehr unschön ist.

Das anfängliche IApp Interface wurde folgendermassen implementiert:



Abbildung - Anfängliche Implementation des Interfaces IApp

Wie in Abbildung 3 - Anfängliche Implementation des Interfaces IApp erkennbar ist, ist das Interface relativ gross und stellt verschiedenste Services zur Verfügung, die nichts miteinander zu tun haben. Beispielsweise das ResourceDirectory Property, welches die Extension-Dateien zur Verfügung stellt oder der SkeletonChangedEvent, der vom Framework aufgerufen werden soll, sobald sich das Skelett verändert hat.

Ändert sich dieses Interface, beispielsweise durch Hinzufügen neuer Funktionalität, müssen auch immer alle Extensions neu kompiliert werden. Dies ist suboptimal, speziell dann, wenn die Extensions von verschiedenen Personen gewartet werden.

##### Lösung

Die Lösung ist an das Extension Interface (<http://wiki.hsr.ch/APF/files/ExtensionInterface.pdf> TODO) angelehnt. Es bietet einen Ansatz, das Problem des ständig ändernden Interfaces zu lösen. In der Video Wall Applikation wurde das Extension Interface in abgeänderter Form angewendet, ohne die Vererbung des Root Interfaces. Zusätzlich wurde Unity <http://unity.codeplex.com/> TODO verwendet um die Factory aus dem Extension Interface zu ersetzen. Die gegenwärtige Implementation des Interfaces sieht folgendermassen aus:



Abbildung - Das IApp Interface

Das IApp Interface (siehe Abbildung 4 - Das IApp Interface) bietet einen Einstiegspunkt. Da jede Applikation dieses Interface implementiert, sind hier nur die Anforderungen beschrieben, die jede Applikation anbieten muss. Speziell ist die Methode Activate, die auf jeder vom Framework zu ladenden Extension genau einmal aufgerufen wird. In dieser Methode kann die Extension über das IVideoWallServiceProvider-Objekt weitere Services anfordern:



Abbildung - Durch den IVideoWallServiceProvider können weitere Extensions geladen werden

Über die Methode GetExtension des IVideoWallServiceProviders aus obiger Abbildung kann die Extension weitere Services (IVideoWallService) anfordern.

Das IVideoWallService Interface ist ein Marker-Interface. Es ist nicht vorgesehen, dass Applikationen weitere Extensions registrieren können. Dies stellt den Hauptunterschied zum Extension Interface Pattern dar.

###### Dynamische Sicht

Nachfolgend ein Sequenzdiagramm, welches den Ablauf, wie die Applikationen vom Framework (Video Wall Applikation) geladen und aktiviert werden, veranschaulicht.



Abbildung - Sequenzdiagramm, Ablauf des Ladens und Aktivierens von Applikationen durch das Framework

### Usability Tests

#### Test 1: Wizard of Oz

Am 27. März 2012 wurde der Test (für die Erarbeitung siehe TODO link Domain Analyse) durchgeführt. Bei diesem galt es, die Hypothese „Meine Hand ist die Maus“ zu bestätigen. Um dies zu prüfen, wurde mithilfe einer WPF Applikation ein Wizard of Oz Experiment durchgeführt. Die Testpersonen wurden gebeten, laut mitzudenken.



Abbildung - Testdurchführung Wizard of Oz mit einem Probanden

Der Test wurde mit sieben Personen durchgeführt, welche das Testszenario (siehe I.2.1.2.1 Testszenario) durchspielten. Alle Testpersonen konnten die Aufgabe ohne grosse Probleme lösen. Die während des Tests gemachten Notizen befinden sich im Anhang (TODO).

Nachfolgend ist beschrieben, was die Applikation, welche beim Test eingesetzt wurde, kann und wie sie gesteuert wird.



Abbildung - Testapplikation

Die Testapplikation besteht aus zwei Ansichten. In der einen können Poster gelesen werden, in der anderen Ansicht wird das Mittagsmenü der Mensa angezeigt.

Die blauen Punkte in der Abbildung 6 - Testapplikation dienen der Beschriftung der einzelnen Komponenten in der Poster-Ansicht:

1. Das Menu. Hier kann zwischen den Ansichten (hier Poster und Mittagsmenü) gewechselt werden.
2. Der Navigationspfeil nach links. Er wird dazu benutzt, um nach links zum vorhergehenden Poster zu navigieren.
3. Der Navigationspfeil nach rechts. Er wird dazu benutzt, um nach rechts zum nachfolgenden Poster zu navigieren.
4. Die Hand. Sie symbolisiert die Hand der Testperson und befindet sich dort, wo die Testperson hinzeigt. Die Mauszeiger-Hand wird am Computer von den Testüberwachern bewegt, und zwar synchron zu den Bewegungen der Hand der Testperson.
5. Das Skelett der Testperson. Es dient dazu, der Testperson zu zeigen, dass sie erkannt wird und merkt, dass sie durch Körperbewegungen die Applikation steuern kann. Das Skelett wird mithilfe von Kinect angezeigt.

Damit eine Schaltfläche effektiv gedrückt wird, muss die Testperson ihre Hand eine Weile darüber halten. Dabei wir über der Mauszeiger-Hand ein Uhr-Symbol angezeigt. Dies dient der Testperson als Feedback, damit diese weiss, dass die Applikation die Geste erkannt hat.

##### Testszenario

Du bist Student/in an der HSR und warst heute Morgen von 8 bis 10 Uhr in einer Vorlesung. Es ist nun Pause und du gehst gerade in die Mensa, um ein Brötchen zu kaufen. Dabei fällt dir die grosse Monitorwand im Eingangsbereich des Gebäudes 4 auf. Du gehst auf die Wand zu.

###### Aufgabe

Du stehst nun also vor der grossen Monitorwand (hier im Test ist das die Projektion des Beamers). Du bist neugierig und möchtest herausfinden, was die Video Wall alles für Funktionen bietet.

##### Resultat

Die Beobachtungen und Notizen, welche während der Durchführung des Tests gemacht wurden, sind in der nachfolgenden Tabelle zusammengefasst:

|  |  |  |  |
| --- | --- | --- | --- |
| Testperson kam insgesamt ... zurecht. | 3 x sehr gut | 4 x gut |  |
| Testperson hatte Schwierigkeiten bei der Bearbeitung der Aufgabe. | 7 x gar nicht |  |  |
| Testperson zögerte bei der Bearbeitung der Aufgabe. | 5 x gar nicht | 1 x mittelmässig | 1 x ziemlich |
| Testperson war langsam bei der Bearbeitung der Aufgabe. | 6 x gar nicht | 1 x kaum |  |
| Testperson positionierte sich von Anfang an korrekt. | Sechs von sieben Testpersonen positionierten sich von Anfang an mit dem richtigen Abstand zur Wand und dem Kinect. | | |
| Testperson merkte ..., dass das Skelett ihre Bewegungen imitiert. | 5 x ausserordentlich schnell | 1 x ziemlich schnell | 1 x fast bis zum Schluss nicht |

Tabelle - Zusammenfassung Resultat empirischer formativer Test

Weitere Beobachtungen:

* Vier Testpersonen wollten die Schaltfläche (Pfeil oder Menu-Button) mit einer Bewegung der Hand nach vorne oder durch das machen einer Faust betätigen.
* Vier Testpersonen hätten gerne das Poster mittels Zoomgeste vergrössert.
* Vier Testpersonen wollten die Bilder auf den Postern oder das Poster insgesamt anklicken.
* Zwei Testpersonen wollten auch mit der linken Hand steuern.
* Zwei Testpersonen wollten mit einer Wischgeste zum nächsten Poster übergehen.

Weiter merkten die Testpersonen an, dass:

* Sie sich auch vorstellen kann, dass das Poster grösser wird, wenn er näher zur Wand geht.
* Sie sich vorstellen kann, auch mit Doppelklick oder über Zoomleiste (Slider) zu zoomen.

Das Fazit des Tests ist, dass die Hypothese „Meine Hand ist die Maus“ bestätigt werden konnte. Bei diesem Test wurde die Applikation nicht mit mehr als sieben Personen geprüft. Dies, da die Resultate so deutlich waren, dass eine statistische Analyse nicht für nötig befunden wurde. Aufgrund dieses Resultats wird die Applikation so weiterentwickelt, dass die Video Wall nicht mit Gesten sondern nur mit der Hand gesteuert wird. Zusätzlich konnte auch das GUI verifiziert werden. Für die Testpersonen war sehr schnell klar, für was die Pfeile und das Menu verwendet werden können.

#### Test 2: Reaktion der Nutzer

Nachdem die gewünschte Steuerung der Wall über die Hand implementiert wurde, entschied sich das Team dazu, deren Eignung am 20.04.2012 nochmals zu testen. Zudem sollte beobachtet werden, wie Passanten des Gebäudes 4 auf die Video Wall reagieren.

Der Test wurde im Eingangsbereich des Gebäudes 4 aufgestellt. Da sich an der Wand, an welcher die Video Wall installiert werden soll, zurzeit noch ein Infostand befindet, wurden die gegenüberliegende Wand genutzt. Um die Video Wall mit einfachen Mitteln nachstellen zu können, wurde ein Kurzdistanzbeamer installiert, welcher die Applikation, welche von einem Laptop aus gestartet wurde, an die Wand projizierte. Kinect konnte nicht direkt unterhalb der Projektion platziert werden, da sonst der Kurzdistanzbeamer genau im Interaktionsbereich der Applikation gelegen und eine Bedienung durch den Nutzer verunmöglicht hätte. Es wurde daher entschieden, den Sensor in den Bereich zwischen der Wand und dem Beamer, leicht hinter den Beamer versetzt, zu stellen. Somit ergab sich zwischen dem Sensor und der Zone, durch welche die meisten Passanten auf dem Weg in die Mensa gehen, ein optimaler Erkennungsabstand von 3-4 Metern.



Abbildung - Test 2: Reaktion der Nutzer

Schon das Aufstellen der Geräte im Gebäude 4 zog grosse Aufmerksamkeit auf sich. Als die Applikation schliesslich gestartet war, wurde sie von praktisch allen Passanten registriert. Es liessen sich jedoch nicht alle dazu animieren, anzuhalten und die Applikation genauer zu betrachten. Dies könnte durch einen Teaser wesentlich verbessert werden. Die meisten Passanten wurden erst neugierig auf die Applikation, als ihr Skelett im Vorbeigehen im unteren Bildschirmbereich auftauchte.

Die Steuerung mit der Hand wurde von den meisten schnell verstanden, einige wenige begnügten sich damit, einige Verrenkungen zu machen und zuzuschauen, wie das Skelett diese nachmacht.  
Obwohl die Handsteuerung bei kleinen Tests in der Testumgebung des Bachelorzimmers ohne grosse Probleme funktionierte, zuckte der Handpointer bei diesem Usability Test merklich. Dies führte teilweise dazu, dass einige Benutzer schnell das Interesse an der Applikation verloren, da diese so schwierig zu bedienen war. Die Verbesserung der Steuerung wurde daher als wichtiger nächster Punkt bei den bevorstehenden Tätigkeiten des Teams aufgeführt und als User Story erfasst.

#### Test 3: Reaktion auf Demomodus



Abbildung - Test 3: Reaktion auf Demomodus

Nachdem der Demomodus implementiert war, sollte auch dieser wieder von den Nutzern geprüft werden. Daher testete das Team den Demomodus am 18.05.2012. Der Aufbau verlief wieder gleich wie schon bei I.2.2 Test 2: Reaktion der Nutzer.

Auch dieses Mal liessen sich nicht alle Personen dazu animieren, vor der Wall stehen zu bleiben. Vor allem Einzelpersonen widmeten dem Testaufbau höchstens einen kurzen Blick, gingen aber zielgerichtet daran vorbei. Gruppen blieben hingegen eher stehen. Ein Gruppenmitglied interagierte mit der Wall und die anderen schauten zu. Auch nach dem der Demomodus die Applikation komplettiert, ist das Skelett die Attraktion. Die Benutzer verrenkten sich vor der Wall, um zu sehen, wie das Skelett diese Bewegungen nachahmt.

Folgende Nachteile der aktuellen Implementation des Demomodus wurden durch diesen Test ersichtlich:

* Sobald eine Person im Demomodus erkannt wird, verschwindet der Teaser-Text und ein Timer wird gestartet, der Countdown wird angezeigt. Es dauert zu lange, bis der Timer aktiviert wird, da der Abfragerhythmus zur Erkennung von Passanten noch nicht optimal ist. Daher muss das Intervall zwischen zwei Skelettabfragen verkürzt werden. Ansonsten haben die Personen die Wall schon passiert, bevor eine Reaktion der Wall auf die Erkennung ihres Skelettes ersichtlich ist.
* Das Skelett sollte am besten schon beim Herunterzählen des Countdowns (also noch im Demomodus) angezeigt werden, damit dem Nutzer klar ist, dass er erkannt wurde.
* Bei der Poster-App muss der Teaser-Text überdacht werden. Dieser lautete „Willst du etwas lernen?“ – was von einigen Passanten im Vorbeilaufen lautstark mit „Nein!“ beantwortet wurde.

Nach der Behebung dieser Nachteile wird die Applikation wieder einem Usability Test unterzogen.

#### Test 4: Grafisches Design

Mit diesem Usability Test sollte am 22.05.2012 getestet werden, ob das grafische Design verständlich ist. Weiter wurde sichergestellt, dass sich die Verbesserungen am Demomodus, welche nach dem vorhergehenden Usability Test (siehe Unterkapitel I.1.3.3 Test 3: Reaktion auf Demomodus) vorgenommen wurden, bewähren. Der Aufbau verlief wieder gleich wie schon bei I.2.2 Test 2: Reaktion der Nutzer.



Abbildung - Test 4: Grafisches Design

Folgende Beobachtungen konnten während der Durchführung des Tests gemacht werden:

* Die Tabs im Menu sind noch nicht deutlich als Tabs ersichtlich, weshalb einige Nutzer nicht wussten, wo sie klicken können.
* Der Hand Cursor soll sich nicht drehen, wenn er sich im Menu auf dem Tab befindet, welches bereits aktiv ist. Einige Benutzer versuchten daher zu den Postern zu wechseln, obwohl diese Applikation bereits aktiv war.
* Einige Nutzer versuchten, das Mittagsmenü oder Elemente auf den Postern anzuklicken. Der Hand Cursor soll, je nachdem ob er sich über einem interaktiven Objekt (z.B. ein Button) befindet oder nicht, anders gekennzeichnet sein. Beispielsweise soll die Hand durchgestrichen sein oder das Bild soll mehr Transparenz haben.

Der erste Punkt wird noch im Rahmen dieser Arbeit umgesetzt. Die anderen zwei Beobachtungen wurden als User Stories in den Backlog aufgenommen (TODO Referenz Backlog).

#### Systemtests

Die Systemtests orientieren sich an den definierten User Stories, die dann im entsprechenden Sprint umgesetzt wurden.

##### Sprint 7

Testperson: Delia Treichler A = im Architekturprototypen enthalten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket# | Titel | Beschreibung | Resultat | Datum |
| A | Poster werden angezeigt | Es wird das aktuelle Poster angezeigt. | Ok | 16.04.2012 |
| A | Poster browsen | Es kann zum nächsten und zum vorhergehenden Poster gewechselt werden. | Ok | 16.04.2012 |
| A | Hand Cursor wird dargestellt | Der Hand Cursor wird als blauer Punkt dargestellt. | Ok | 16.04.2012 |
| A | Eigenes Skelett wird dargestellt | Die verschiedenen Skelettjoints des Benutzers werden angezeigt. | Ok | 16.04.2012 |
| A | Sofortiges Erfolgserlebnis für Einstieg sichergestellt | Das sofortige Erfolgserlebnis ist durch das Skelett sichergestellt. | Ok | 16.04.2012 |

##### Sprint 8

Testperson: Delia Treichler A = im Architekturprototypen enthalten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket# | Titel | Beschreibung | Resultat | Datum |
| A | Poster werden angezeigt | Es wird das aktuelle Poster angezeigt. | Ok | 23.04.2012 |
| A | Poster browsen | Es kann zum nächsten und zum vorhergehenden Poster gewechselt werden. | Ok | 23.04.2012 |
| A | Hand Cursor wird dargestellt | Der Hand Cursor wird als blauer Punkt dargestellt. | Ok | 16.04.2012 |
| A | Eigenes Skelett wird dargestellt | Die verschiedenen Skelettjoints des Benutzers werden angezeigt. | Ok | 16.04.2012 |
| A | Sofortiges Erfolgserlebnis für Einstieg sichergestellt | Das sofortige Erfolgserlebnis ist durch das Skelett sichergestellt. | Ok | 23.04.2012 |
| 769 | Pointer für die Hand schön dargestellt | Der Hand Cursor wird als rechte Hand dargestellt. | Ok | 23.04.2012 |
| 779 | Skelett schön dargestellt | Das Skelett wird als Strichmännlein dargestellt. | Ok | 23.04.2012 |
| 870 | Hand Cursor ruckelt weniger 1 | Der Hand Cursor zittert nicht mehr so fest. | Ok | 23.04.2012 |

##### Sprint 9

Testperson: Delia Treichler A = im Architekturprototypen enthalten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket# | Titel | Beschreibung | Resultat | Datum |
| A | Poster werden angezeigt | Es wird das aktuelle Poster angezeigt. | Ok | 27.04.2012 |
| A | Poster browsen | Es kann zum nächsten und zum vorhergehenden Poster gewechselt werden. | Ok | 27.04.2012 |
| A | Hand Cursor wird dargestellt | Der Hand Cursor wird als blauer Punkt dargestellt. | Ok | 16.04.2012 |
| A | Eigenes Skelett wird dargestellt | Die verschiedenen Skelettjoints des Benutzers werden angezeigt. | Ok | 16.04.2012 |
| A | Sofortiges Erfolgserlebnis für Einstieg sichergestellt | Das sofortige Erfolgserlebnis ist durch das Skelett sichergestellt. | Ok | 27.04.2012 |
| 769 | Pointer für die Hand schön dargestellt | Der Hand Cursor wird als rechte Hand dargestellt. | Ok | 27.04.2012 |
| 779 | Skelett schön dargestellt | Das Skelett wird als Strichmännlein dargestellt. | Ok | 27.04.2012 |
| 870 | Hand Cursor ruckelt weniger 1 | Der Hand Cursor zittert nicht mehr so fest. | Ok | 27.04.2012 |
| 786 | Video wird dargestellt | Das Video wird in WPF dargestellt. | Ok | 27.04.2012 |
| 785 | Applikation ist mit linker Hand bedienbar | Die Applikation kann sowohl mit der rechten als auch der linken Hand bedient werden. | Ok | 27.04.2012 |

##### Sprint 10

Testperson: Delia Treichler A = im Architekturprototypen enthalten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket# | Titel | Beschreibung | Resultat | Datum |
| A | Poster werden angezeigt | Es wird das aktuelle Poster angezeigt. | Ok | 07.05.2012 |
| A | Poster browsen | Es kann zum nächsten und zum vorhergehenden Poster gewechselt werden. | Ok | 07.05.2012 |
| A | Hand Cursor wird dargestellt | Der Hand Cursor wird als blauer Punkt dargestellt. | Ok | 16.04.2012 |
| A | Eigenes Skelett wird dargestellt | Die verschiedenen Skelettjoints des Benutzers werden angezeigt. | Ok | 16.04.2012 |
| A | Sofortiges Erfolgserlebnis für Einstieg sichergestellt | Das sofortige Erfolgserlebnis ist durch das Skelett sichergestellt. | Ok | 07.05.2012 |
| 769 | Pointer für die Hand schön dargestellt | Der Hand Cursor wird als rechte Hand dargestellt. | Ok | 07.05.2012 |
| 779 | Skelett schön dargestellt | Das Skelett wird als Strichmännlein dargestellt. | Ok | 07.05.2012 |
| 870 | Hand Cursor ruckelt weniger 1 | Der Hand Cursor zittert nicht mehr so fest. | Ok | 07.05.2012 |
| 786 | Video wird dargestellt | Das Video wird in WPF dargestellt. | Ok | 07.05.2012 |
| 785 | Applikation ist mit linker Hand bedienbar | Die Applikation kann sowohl mit der rechten als auch der linken Hand bedient werden. | Ok | 07.05.2012 |
| 798 | Plug-in Möglichkeit | Ein Plug-in kann automatisch in die Main-Applikation geladen werden. | Ok | 07.05.2012 |

##### Sprint 11

Testperson: Delia Treichler A = im Architekturprototypen enthalten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket# | Titel | Beschreibung | Resultat | Datum |
| A | Poster werden angezeigt | Es wird das aktuelle Poster angezeigt. | Ok | 14.05.2012 |
| A | Poster browsen | Es kann zum nächsten und zum vorhergehenden Poster gewechselt werden. | Ok | 14.05.2012 |
| A | Hand Cursor wird dargestellt | Der Hand Cursor wird als blauer Punkt dargestellt. | Ok | 16.04.2012 |
| A | Eigenes Skelett wird dargestellt | Die verschiedenen Skelettjoints des Benutzers werden angezeigt. | Ok | 16.04.2012 |
| A | Sofortiges Erfolgserlebnis für Einstieg sichergestellt | Das sofortige Erfolgserlebnis ist durch das Skelett sichergestellt. | Ok | 14.05.2012 |
| 769 | Pointer für die Hand schön dargestellt | Der Hand Cursor wird als rechte Hand dargestellt. | Ok | 14.05.2012 |
| 779 | Skelett schön dargestellt | Das Skelett wird als Strichmännlein dargestellt. | Ok | 14.05.2012 |
| 870 | Hand Cursor ruckelt weniger 1 | Der Hand Cursor zittert nicht mehr so fest. | Ok | 14.05.2012 |
| 786 | Video wird dargestellt | Das Video wird in WPF dargestellt. | Ok | 14.05.2012 |
| 785 | Applikation ist mit linker Hand bedienbar | Die Applikation kann sowohl mit der rechten als auch der linken Hand bedient werden. | Ok | 14.05.2012 |
| 798 | Plug-in Möglichkeit | Ein Plugin kann automatisch in die Main-Applikation geladen werden. | Ok | 14.05.2012 |
| 833 | Demomodus: Vom Demomodus wird zum Interaktionsmodus gewechselt | Wenn die Applikation im Demomodus ist und ich sie bedienen möchte (Skeletterkennung), wechselt sie automatisch in den Interaktionsmodus. | Ok | 14.05.2012 |
| 834 | Demomodus: Vom Interaktionsmodus wird zum Demomodus gewechselt | Wenn die Applikation im Interaktionsmodus ist und niemand die Applikation bedient (Skeletterkennung), so wechselt sie automatisch in den Demomodus. | Ok | 14.05.2012 |
| 799 | Bild der Hand ist auf die rechte bzw. linke Hand abgestimmt | Bediene ich die Applikation mit der rechten Hand, so wird der Cursor als rechte Hand dargestellt. Bediene ich die Applikation mit der linken Hand, so ist das Bild des Cursors eine linke Hand. | Ok | 14.05.2012 |

##### Sprint 12

Testperson: Delia Treichler A = im Architekturprototypen enthalten

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ticket# | Titel | Beschreibung | Resultat | Datum |
| A | Poster werden angezeigt | Es wird das aktuelle Poster angezeigt. | Ok | 22.05.2012 |
| A | Poster browsen | Es kann zum nächsten und zum vorhergehenden Poster gewechselt werden. | Ok | 22.05.2012 |
| A | Hand Cursor wird dargestellt | Der Hand Cursor wird als blauer Punkt dargestellt. | Ok | 16.04.2012 |
| A | Eigenes Skelett wird dargestellt | Die verschiedenen Skelettjoints des Benutzers werden angezeigt. | Ok | 16.04.2012 |
| A | Sofortiges Erfolgserlebnis für Einstieg sichergestellt | Das sofortige Erfolgserlebnis ist durch das Skelett sichergestellt. | Ok | 22.05.2012 |
| 769 | Pointer für die Hand schön dargestellt | Der Hand Cursor wird als rechte Hand dargestellt. | Ok | 22.05.2012 |
| 779 | Skelett schön dargestellt | Das Skelett wird als Strichmännlein dargestellt. | Ok | 22.05.2012 |
| 870 | Hand Cursor ruckelt weniger 1 | Der Hand Cursor zittert nicht mehr so fest. | Ok | 22.05.2012 |
| 786 | Video wird dargestellt | Das Video wird in WPF dargestellt. | Ok | 22.05.2012 |
| 785 | Applikation ist mit linker Hand bedienbar | Die Applikation kann sowohl mit der rechten als auch der linken Hand bedient werden. | Ok | 22.05.2012 |
| 798 | Plug-in Möglichkeit | Ein Plugin kann automatisch in die Main-Applikation geladen werden. | Ok | 22.05.2012 |
| 800 | Mittagsmenü App in Plugin umgewandelt | Die Mittagsmenü-Applikation besteht als Plugin und kann durch Nr. 14 in der Hauptapplikation angezeigt werden. | Ok | 22.05.2012 |
| 856 | Das Mittagsmenü wird angezeigt. | Das Mittagsmenü wird in der Wochenübersicht angezeigt. | Ok | 22.05.2012 |
| 802 | Poster App in Plugin App umgewandelt | Die Poster-Applikation besteht als Plugin und kann durch Nr. 14 in der Hauptapplikation angezeigt werden. | Ok | 22.05.2012 |
| 833 | Demomodus: Vom Demomodus wird zum Interaktionsmodus gewechselt | Wenn die Applikation im Demomodus ist und ich sie bedienen möchte (Skeletterkennung), wechselt sie automatisch in den Interaktionsmodus. | Ok | 22.05.2012 |
| 834 | Demomodus: Vom Interaktionsmodus wird zum Demomodus gewechselt | Wenn die Applikation im Interaktionsmodus ist und niemand die Applikation bedient (Skeletterkennung), so wechselt sie automatisch in den Demomodus. | Ok | 22.05.2012 |
| 836 | Demomodus: Demotext zu aktiver App wird angezeigt | Wenn die Applikation im Demomodus ist, wird ein attraktiver Teaser-Text angezeigt. | Ok | 22.05.2012 |
| 835 | Demomodus: Apps werden automatisch gewechselt | Wenn die Applikation im Demomodus ist und sich niemand für die Applikation interessiert, so wechselt der Text nach einer definierten Zeit. | Ok | 22.05.2012 |
| 799 | Bild der Hand ist auf die rechte bzw. linke Hand abgestimmt | Bediene ich die Applikation mit der rechten Hand, so wird der Cursor als rechte Hand dargestellt. Bediene ich die Applikation mit der linken Hand, so ist das Bild des Cursors eine linke Hand. | Ok | 22.05.2012 |
| 856 | Mittagsmenü App automatisch aktualisiert | Das Mittagsmenü für den aktuellen Tag wird angezeigt. | Ok | 22.05.2012 |
| 872 | Navigation mit schönen "Tabs" ermöglichen | Die Navigation findet über die Tabs im Menu statt. | Ok | 22.05.2012 |
| 855 | Deployment Entwickler PC möglich | Die Solution kann nach dem SVN-Checkout geöffnet und es kann daran gearbeitet werden. | Ok | 22.05.2012 |

### Code Dokumentation

#### Coding Standards

Die Codings Standards, welche für dieses Projekt gültig sind, wurden mehrheitlich vom Resharper übernommen.

Das exportierte Resharper-Profil kann über den Pfad „code/resharper\_settings/ resharper.DotSettings“ gefunden werden. Nachfolgend wurden die wichtigsten Einstellungen dokumentiert.

##### C# Namenskonventionen

Folgende Namenskonventionen wurden verwendet:

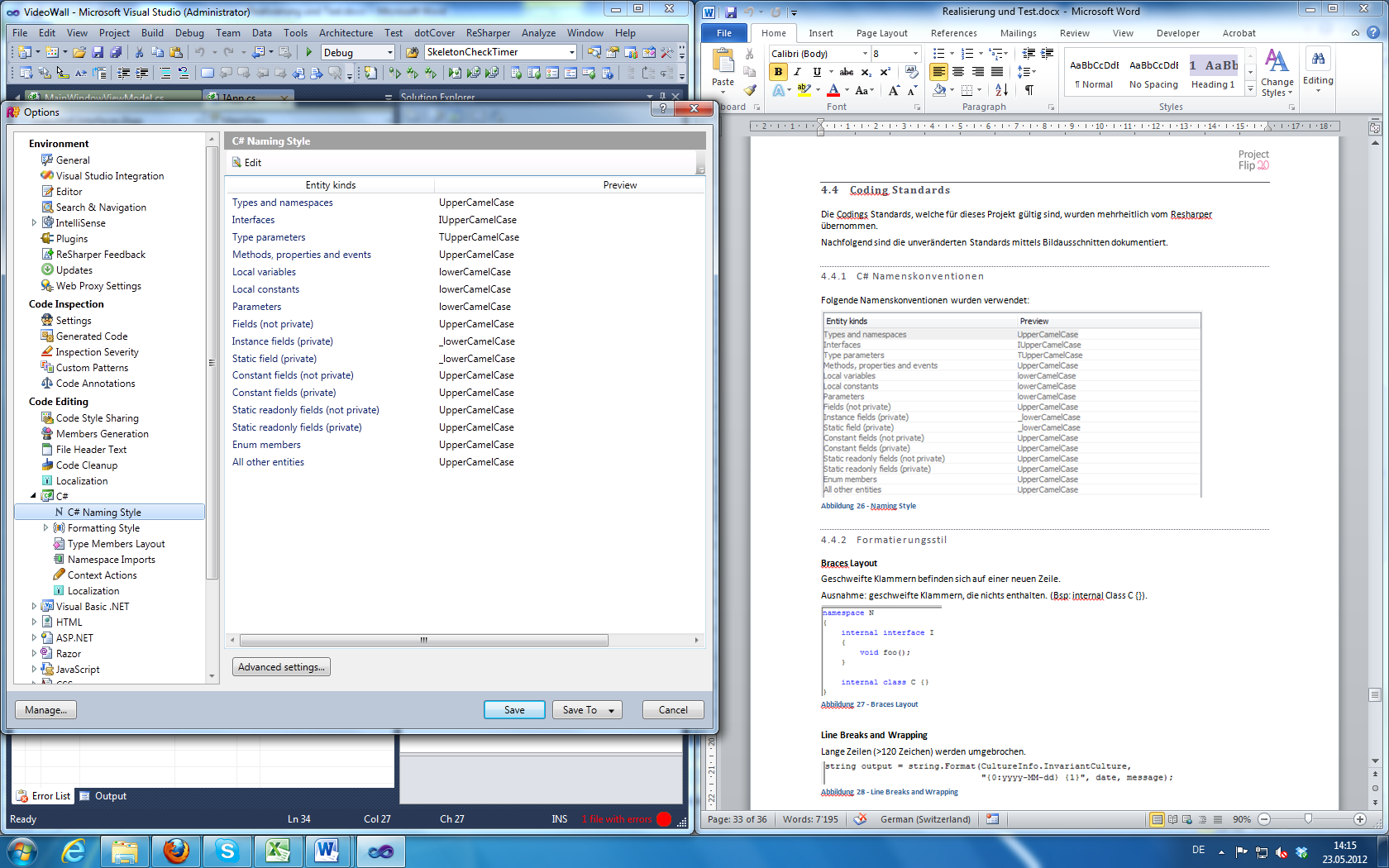


Abbildung - Naming Style

##### Formatierungsstil

**Braces Layout**

Geschweifte Klammer befinden sich auf einer neuen Zeile.

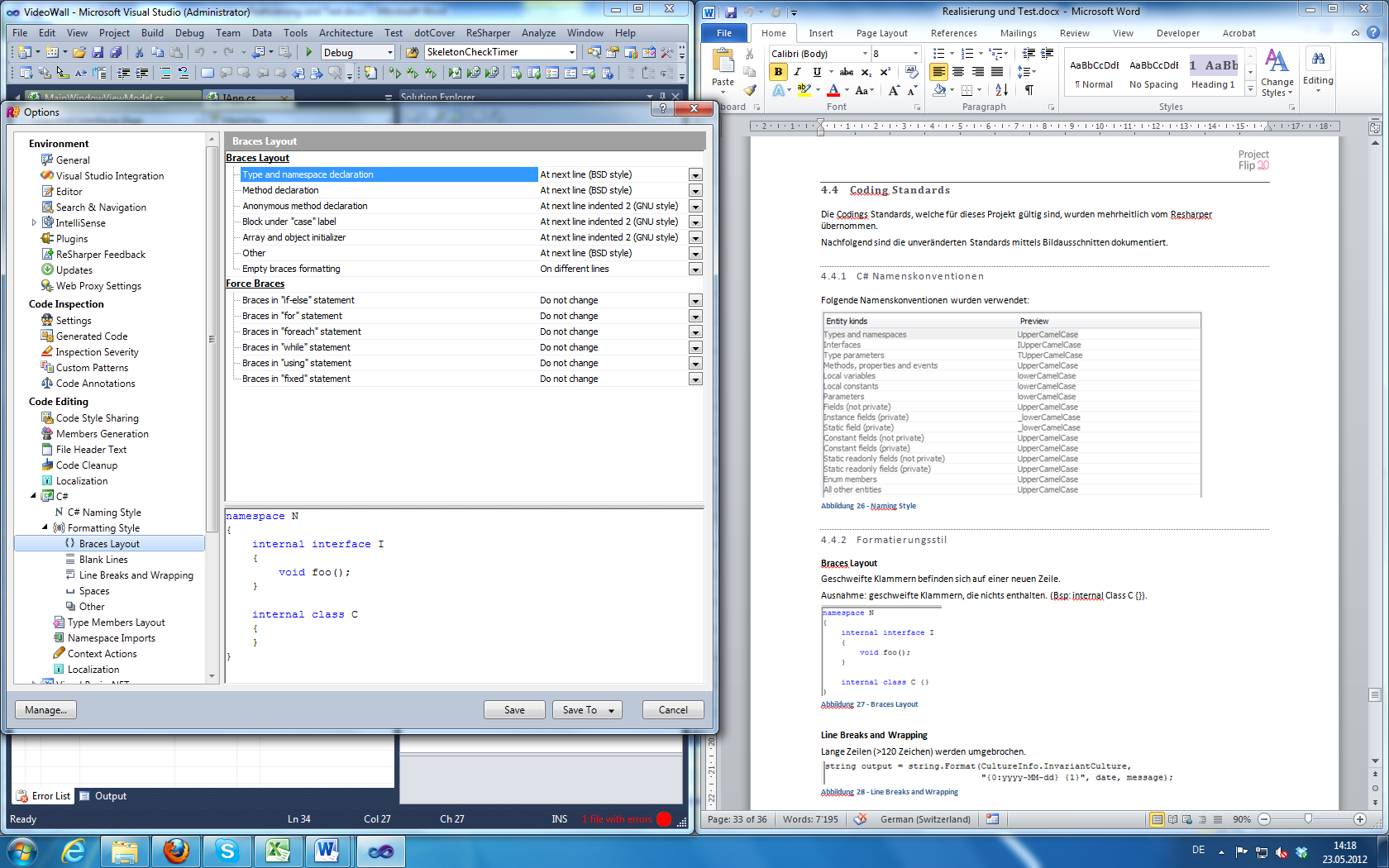


Abbildung 13 – Formatierungsstil, Braces Layout

**Line Breaks and Wrapping**

Lange Zeilen (>120 Zeichen) werden umgebrochen.

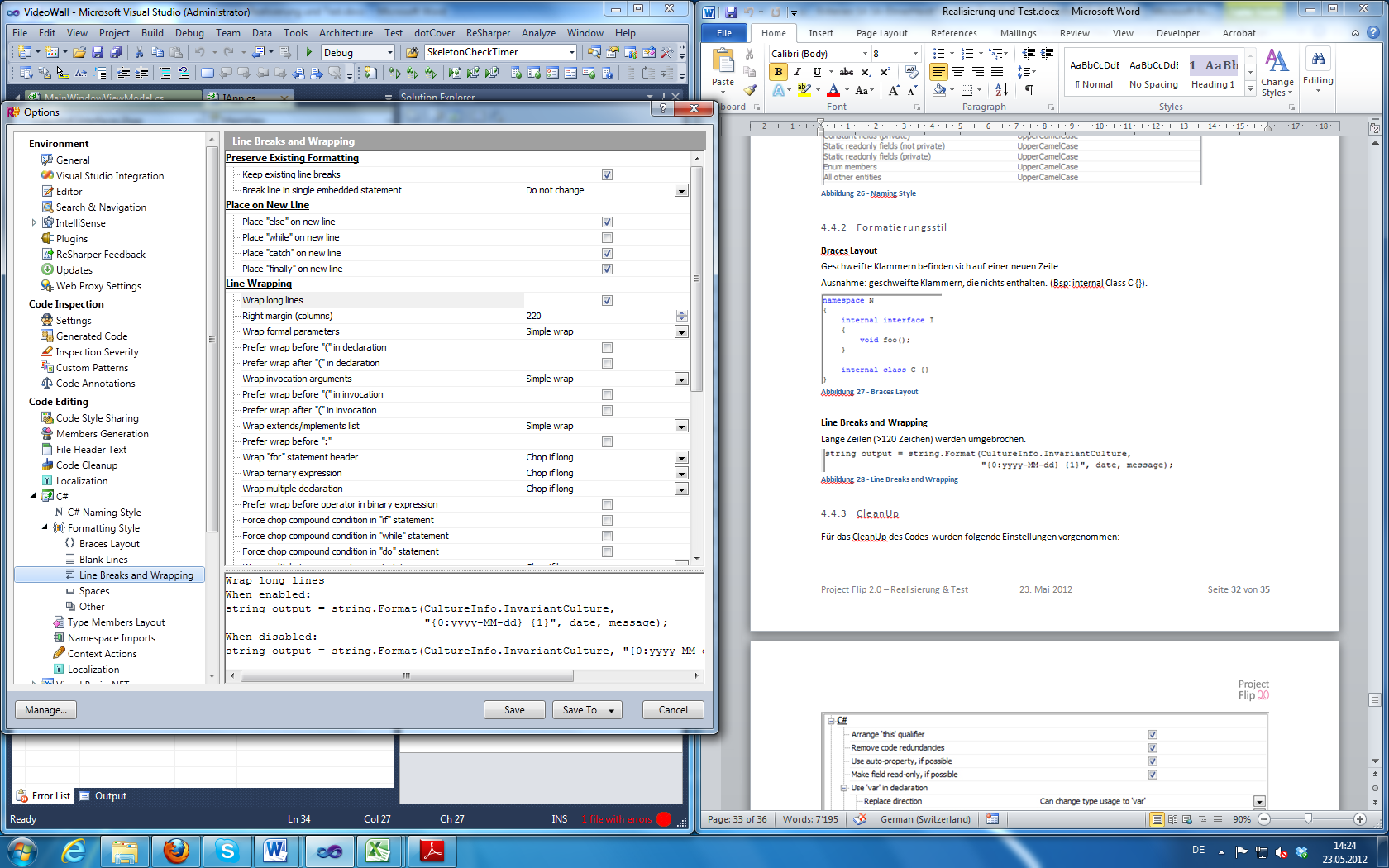


Abbildung - Formatierungsstil, Line Breaks and Wrapping

##### CleanUp

Für das CleanUp des Codes wurden folgende Einstellungen vorgenommen:

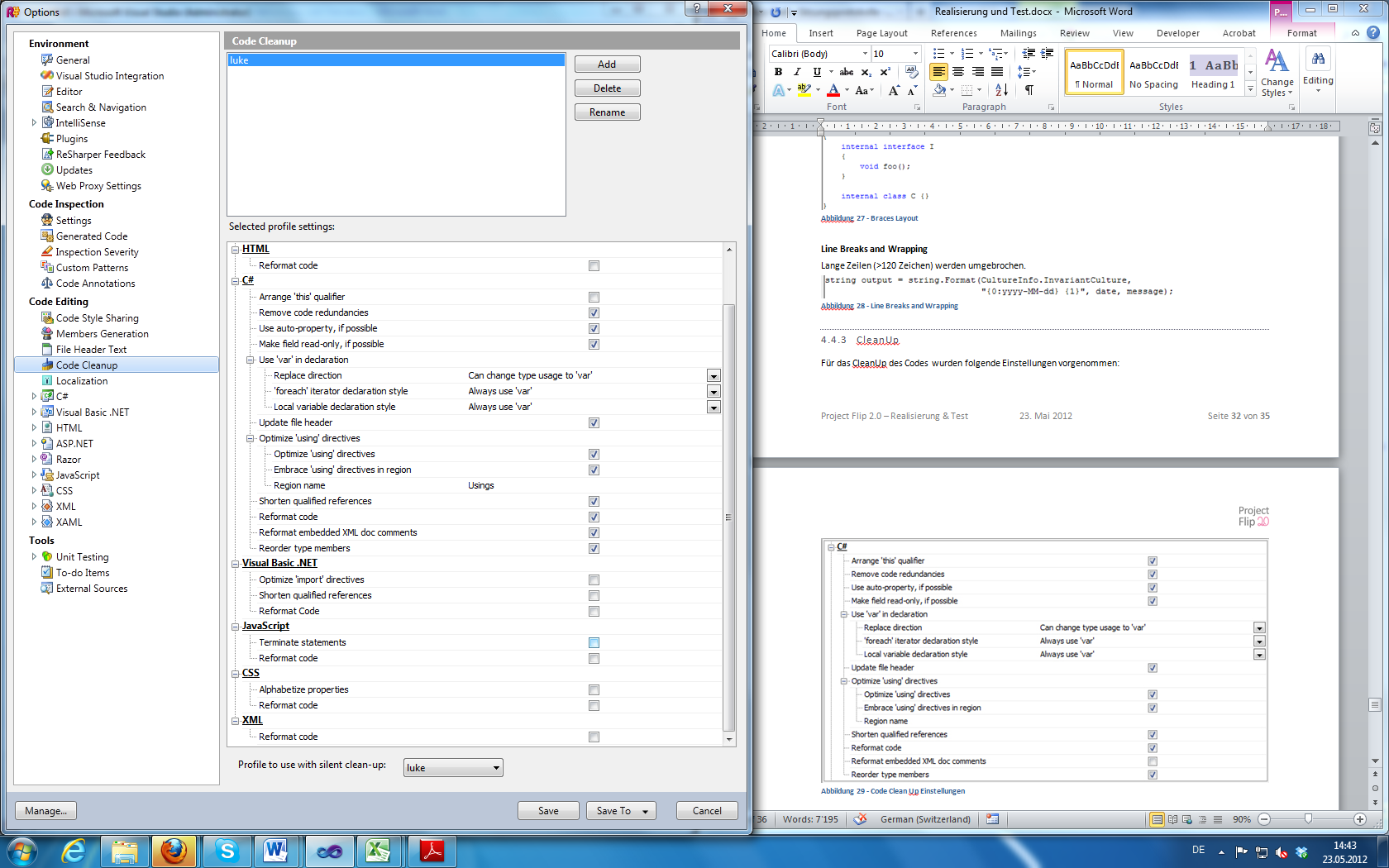


Abbildung - CleanUp Einstellungen

#### Dokumentation Quellcode

Zur Dokumentation des Quellcodes wurde Ghost Doc verwendet. Dadurch war die Dokumentation einiges einfacher, da sinnvolle Standardkommentare generiert werden, die bei Bedarf erweitert werden können. Durch die geschriebenen XML Kommentare ist es auch möglich, eine Dokumentation zu generieren.

Es wurden alle Interfaces, sowie Methoden oder Properties, welche mit Hilfe eines Kommentars besser verstanden werden, kommentiert. Ausgenommen davon sind die Tests und die XAML Dateien. Durch die Kommentare sind der Programmcode und besonders komplexere Methoden für Entwickler leichter verständlich.

1. |  |  |
   | --- | --- |
   | [microsoft12.1] | Microsoft Corporation, „ MEF Community Site”,  <http://mef.codeplex.com/>  letzter Zugriff: 22.05.2012 |

   [↑](#footnote-ref-1)
2. |  |  |
   | --- | --- |
   | [microsoft12.2] | Microsoft Corporation, „Documentation for MEF”,  <http://mef.codeplex.com/documentation>  letzter Zugriff: 22.05.2012 |

   [↑](#footnote-ref-2)