

# QQAT - Objective Quality Assessment Toolkit

PSE - Abschlusspräsentation

Alexander Monev · Artur Eckhart · Georg Ermantraut  
· Johannes Sailer · Sebastian Leidig

Institut für Technische Informatik



Aufgabenstellung

Architektur

Entwicklungsumgebung

Statistiken

Vorführung

Fazit

Aufgabenstellung

Architektur

Entwicklungsumgebung

Statistiken

Vorführung

Fazit

# Aufgabenstellung

- Videos verändern
- Videos analysieren

# Aufgabenstellung

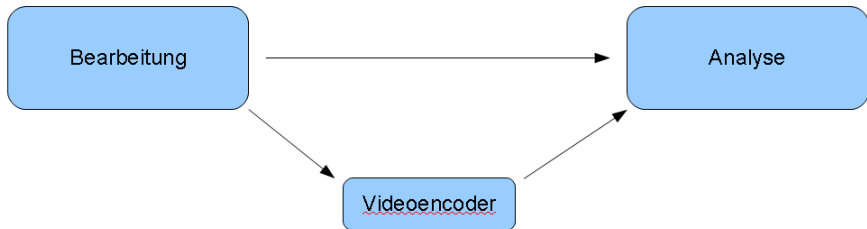
- Videos verändern
- Videos analysieren

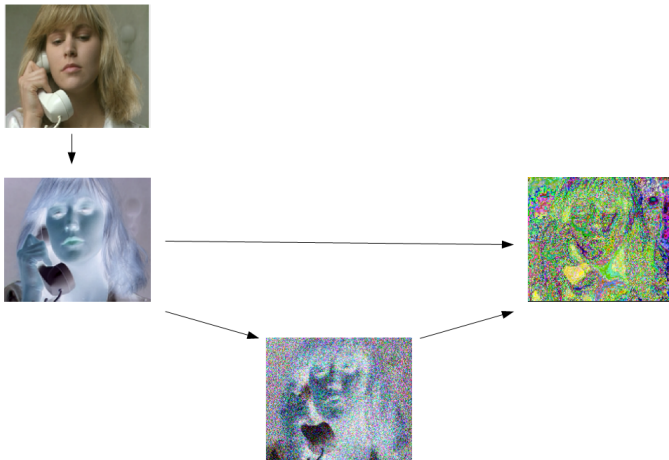
# Aufgabenstellung

- Videos verändern
- Videos analysieren

# Aufgabenstellung

- Videos verändern
- Videos analysieren







Aufgabenstellung

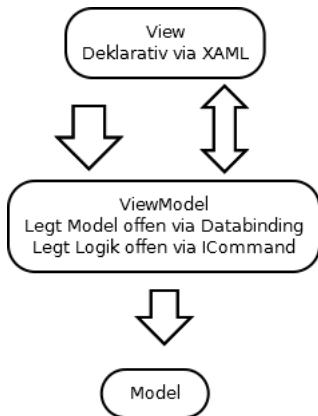
Architektur

Entwicklungsumgebung

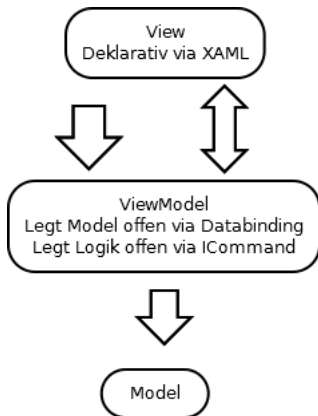
Statistiken

Vorführung

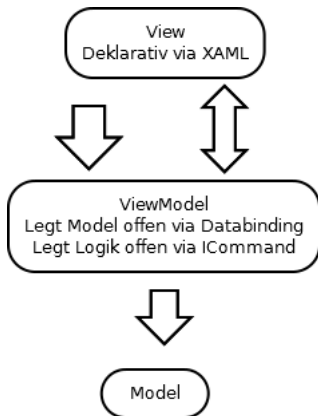
Fazit



- Model View ViewModel Architekturmuster.
- View wurde nahezu vollständig deklarativ definiert.
- Verbesserte Testbarkeit durch Trennung der Darstellung von der Anwendungslogik.

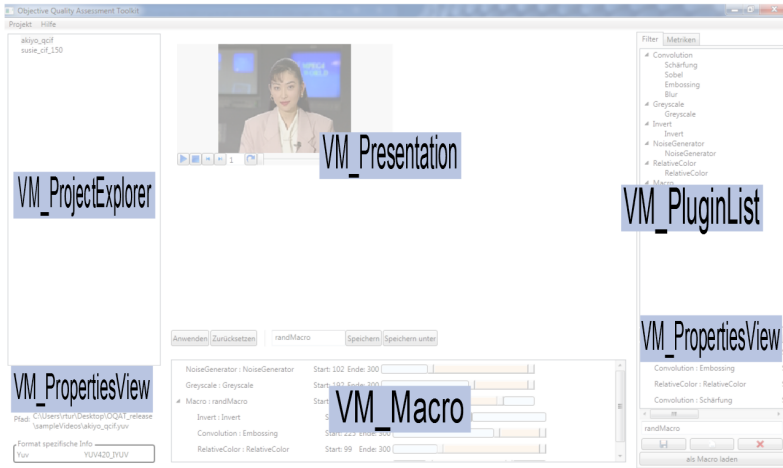


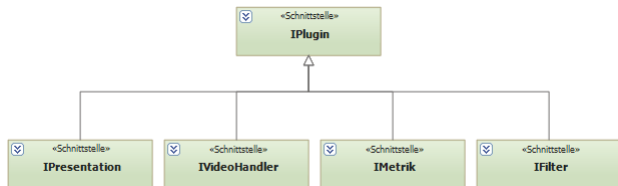
- Model View ViewModel Architekturmuster.
- View wurde nahezu vollständig deklarativ definiert.
- Verbesserte Testbarkeit durch Trennung der Darstellung von der Anwendungslogik.



- Model View ViewModel Architekturmuster.
- View wurde nahezu vollständig deklarativ definiert.
- Verbesserte Testbarkeit durch Trennung der Darstellung von der Anwendungslogik.

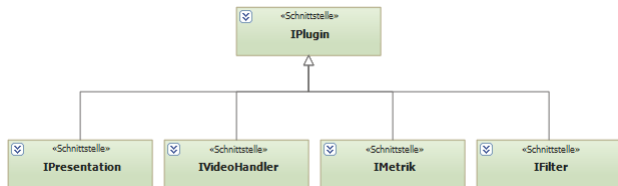
# Von der View zum ViewModel



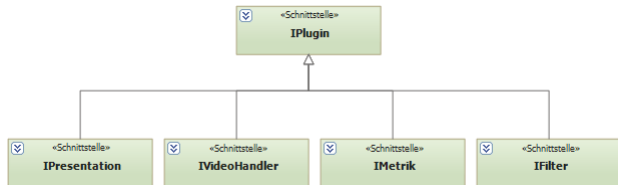


## ■ Managed Extensibility Framework (MEF)

- Plugins werden mit Hilfe exportierter Metadaten und implementierter Schnittstellen identifiziert
- Eine PluginManager Klasse ist für die Koordination und Konsistenz aller Plugins verantwortlich.



- Managed Extensibility Framework (MEF)
- Plugins werden mit Hilfe exportierter Metadaten und implementierter Schnittstellen identifiziert
- Eine PluginManager Klasse ist für die Koordination und Konsistenz aller Plugins verantwortlich.



- Managed Extensibility Framework (MEF)
- Plugins werden mit Hilfe exportierter Metadaten und implementierter Schnittstellen identifiziert
- Eine PluginManager Klasse ist für die Koordination und Konsistenz aller Plugins verantwortlich.



Aufgabenstellung

Architektur

Entwicklungsumgebung

Statistiken

Vorführung

Fazit

# C#



MEF



**AForge.NET**  
FRAMEWORK

Aufgabenstellung

Architektur

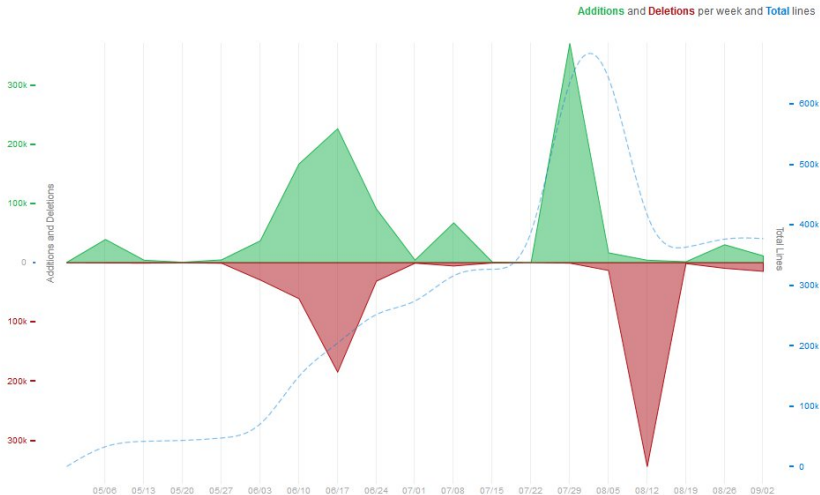
Entwicklungsumgebung

**Statistiken**

Vorführung

Fazit

# Git Code Frequency



<https://github.com/PSE-2012/MMWTV/graphs/code-frequency>

- ~ 14.000 Zeilen, C# Code + XAML + Test + Kommentare
- 63 Klassen
- 86,5 Codeabdeckung
- 82,5 Wartbarkeitsindex (Halstead/Volumen)
- 118 Issues, 843 Commits

- ~ 14.000 Zeilen, C# Code + XAML + Test + Kommentare
- 63 Klassen
- 86,5 Codeabdeckung
- 82,5 Wartbarkeitsindex (Halstead/Volumen)
- 118 Issues, 843 Commits

- ~ 14.000 Zeilen, C# Code + XAML + Test + Kommentare
- 63 Klassen
- 86,5 Codeabdeckung
- 82,5 Wartbarkeitsindex (Halstead/Volumen)
- 118 Issues, 843 Commits

- ~ 14.000 Zeilen, C# Code + XAML + Test + Kommentare
- 63 Klassen
- 86,5 Codeabdeckung
- 82,5 Wartbarkeitsindex (Halstead/Volumen)
- 118 Issues, 843 Commits



- ~ 14.000 Zeilen, C# Code + XAML + Test + Kommentare
- 63 Klassen
- 86,5 Codeabdeckung
- 82,5 Wartbarkeitsindex (Halstead/Volumen)
- 118 Issues, 843 Commits

Aufgabenstellung

Architektur

Entwicklungsumgebung

Statistiken

Vorführung

Fazit

Aufgabenstellung

Architektur

Entwicklungsumgebung

Statistiken

Vorführung

Fazit

- Entwurf simpler machen
- Lange Einarbeitungszeiten
- Mehr Teamarbeit
- Schätzung von Zeitaufwand

# Fazit

## Was haben wir gelernt?

- Entwurf simpler machen
- Lange Einarbeitungszeiten
- Mehr Teamarbeit
- Schätzung von Zeitaufwand

# Fazit

## Was haben wir gelernt?

- Entwurf simpler machen
- Lange Einarbeitungszeiten
- Mehr Teamarbeit
- Schätzung von Zeitaufwand

# Fazit

## Was haben wir gelernt?

- Entwurf simpler machen
- Lange Einarbeitungszeiten
- Mehr Teamarbeit
- Schätzung von Zeitaufwand

Vielen Dank für Ihre Aufmerksamkeit.