



Text Template Transformation Toolkit (T4)

René Leupold

www.dnug-bern.ch

www.dnug-bern.ch/rss.aspx

Follow dnugbe on twitter



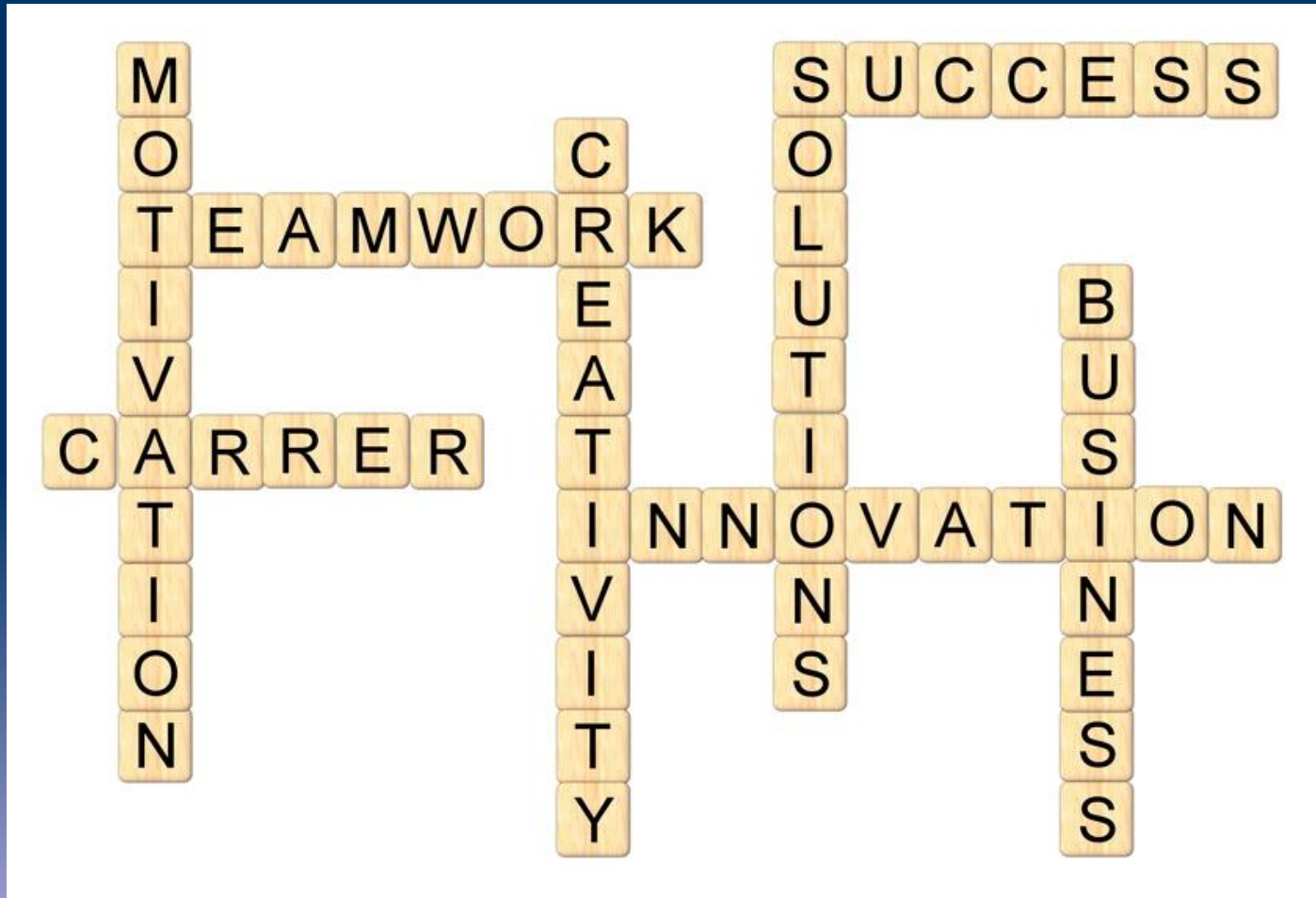
DNUG Bern Sponsoren



Über René Leupold

- connect@databinding.net
- www.databinding.net

Motivation



И

Л
Т

2
2

Agenda

- Prozessablauf und Architektur T4
- Vor- /Nachteile
- Aufbau Text Template
- Anwendungsbeispiele
- Aufbau Preprocessed Template
- Designüberlegungen
- Ressourcen

Prozessablauf

Template

Preprocess



Compiled Template

Execute



Output

```
<#@ template language="C#" #>  
Hello  
<# Write("World!"); #>
```

```
public class GeneratedTextTransformation:  
    Microsoft.VisualStudio.TextTemplating.TextTransformation  
{  
    public override string TransformText()  
    {  
        this.Write("Hello\r\n");  
        Write("World!");  
        return this.GenerationEnvironment.ToString();  
    }  
}
```

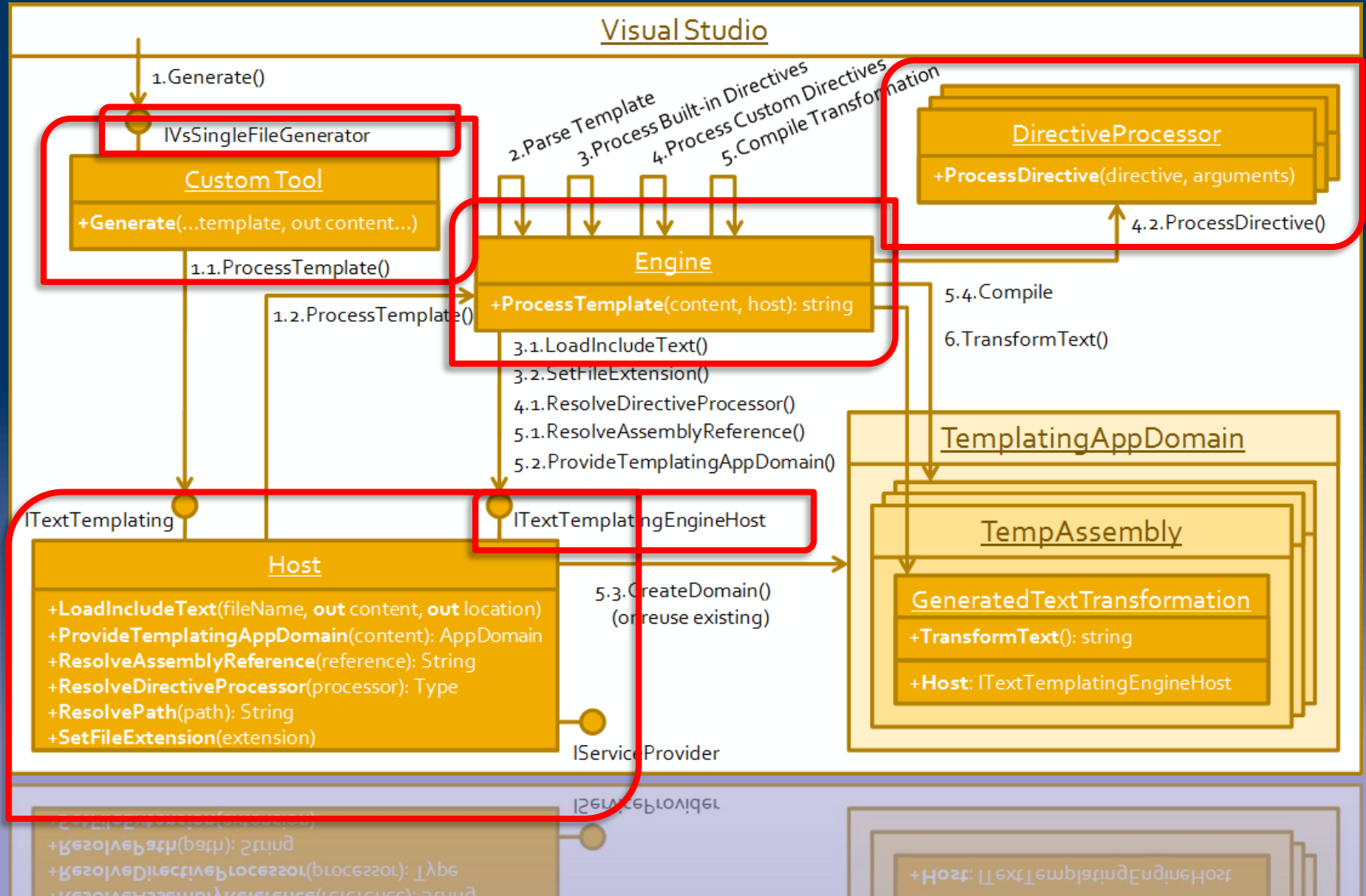
Hello
World!

Output

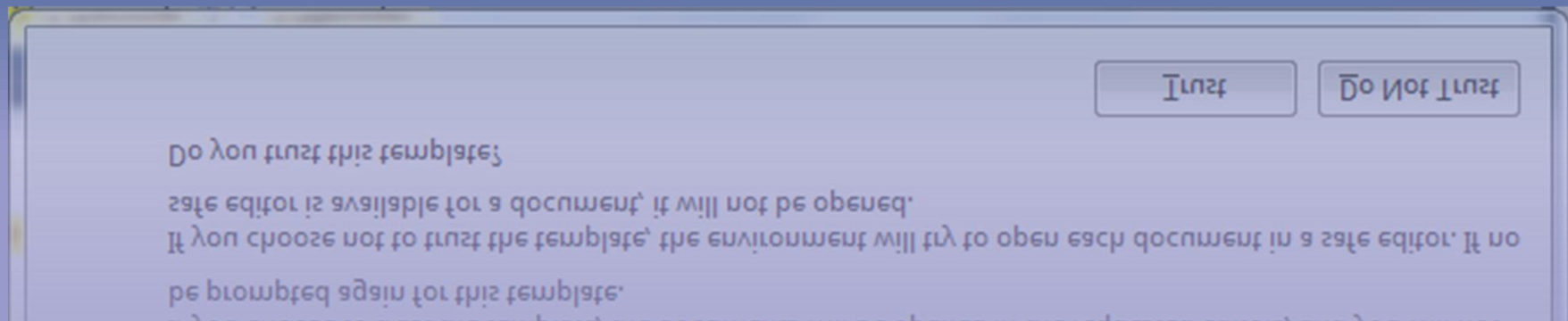
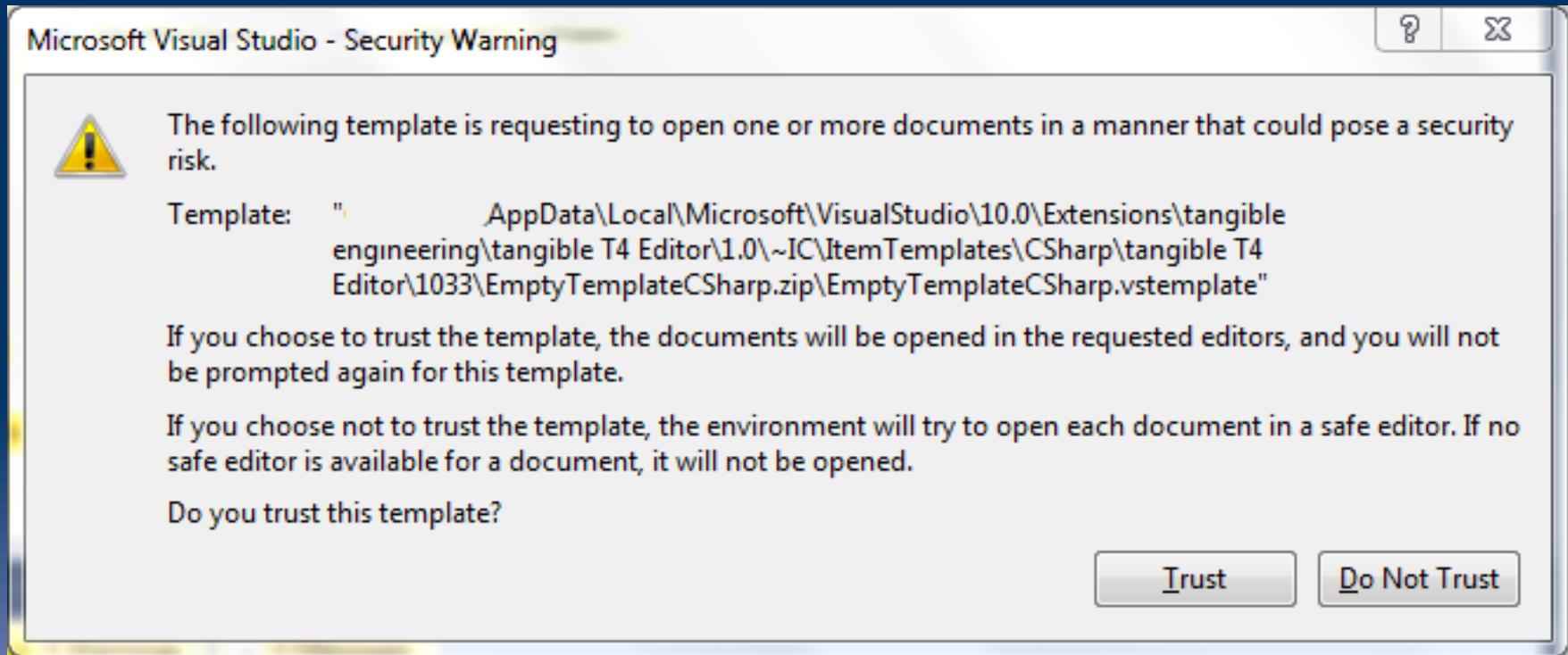
MOUJQ;
H6JJO

Architektur

Visual Studio



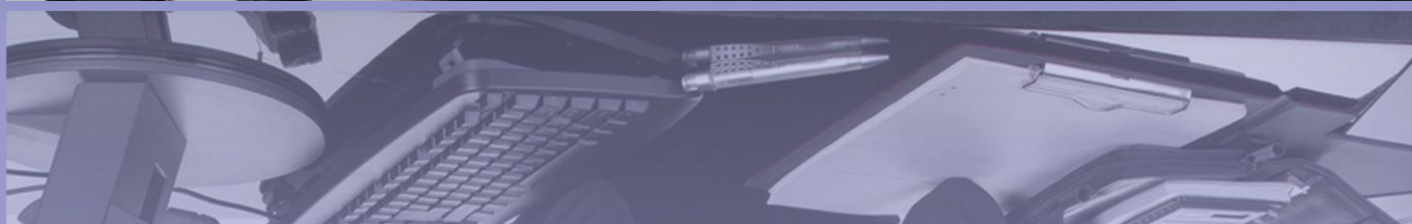
Architektur und Sicherheit



Vorteile



Nachteile



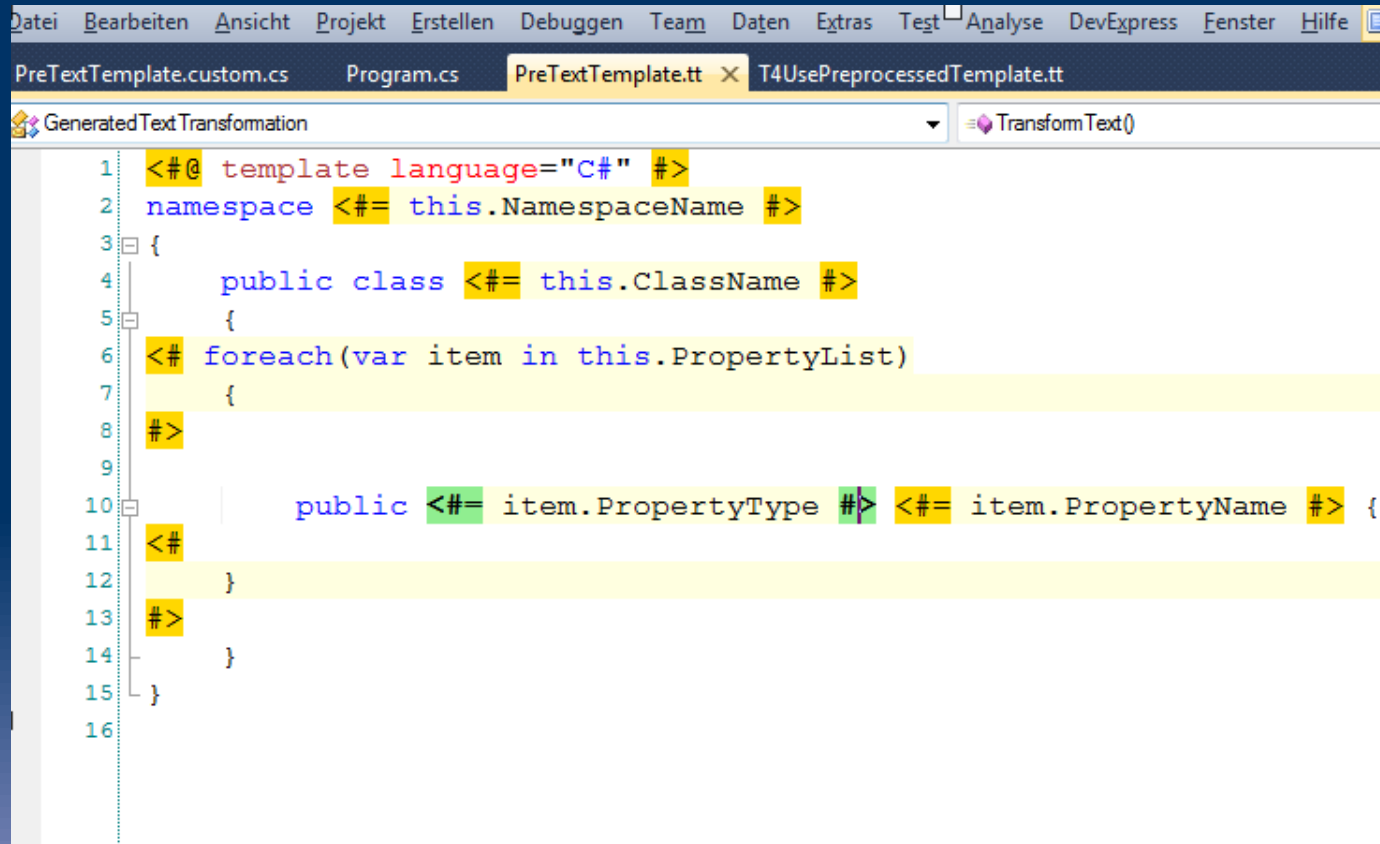
Aufbau Text Template

```
1 <#@ template debug="false" hostspecific="false" language="C#" #>
2 <#@ output extension=".txt" #>
3 <#@ include file="T4DemoArray.ttinclude" #>
4
5 <#
6 string ns = "TestT4";
7 #>
8
9 namespace <#= ns #>
10 {
11 <# foreach (string item in this.GetDemoArray())
12     {
13         this.PushIndent("  ");
14         this.WriteLine(String.Format("public class {0} {{ }}", item));
15         this.PopIndent();
16     }
17 #>
18
19 <# foreach (var item in this.GetDemoArray())
20     {
21
22
23
24
25
26
27
28
29
30     {
31 <# foreach (var item in this.GetDemoArray())
32     #>
33     }
34     this.PopIndent();
35     this.WriteLine(String.Format("public class {0} {{ }}", item));
36     this.PushIndent("  ");
37     {
```

Anwendungsbeispiele

- Anlegen von Prozeduren
- Mehrere Dateien mit VS Automation Model
- Debugging
- Template mit Parametern
- Include-Vorlagen
- Verwenden von Assemblies
- Refactoring
- UML

Aufbau Preprocessed Template



The screenshot shows a Visual Studio window with the following menu items: Datei, Bearbeiten, Ansicht, Projekt, Erstellen, Debuggen, Team, Daten, Extras, Test, Analyse, DevExpress, Fenster, Hilfe. The active window is PreTextTemplate.tt, with other tabs for PreTextTemplate.custom.cs, Program.cs, and T4UsePreprocessedTemplate.tt. The editor displays the following code:

```
1 <#@ template language="C#" #>
2 namespace <#= this.NamespaceName #>
3 {
4     public class <#= this.ClassName #>
5     {
6         <# foreach(var item in this.PropertyList)
7         {
8             #>
9
10            public <#= item.PropertyType #> <#= item.PropertyName #> {
11            <#
12            }
13            #>
14        }
15    }
16
```

```
17 }
18 }
19 #>
20 }
21 <#
```

Designüberlegungen



Ressourcen

- **Gareth Jones** (<http://blogs.msdn.com/b/garethj/>)
- **MSDN** (<http://www.microsoft.com/germany/msdn/webcasts/library.aspx?id=1032456634>)
- **MvcContrib** (<http://mvcccontrib.codeplex.com/>)
- **Oleg Sych** (<http://www.olegsych.com/>)
- **T4 Mind Map** (<http://www.mindmeister.com/de/52082132/t4-textvorlagen>)
- **T4 Toolbox** (<http://t4toolbox.codeplex.com/>)
- **tangible, clarius ...**

Fragen und Diskussion

