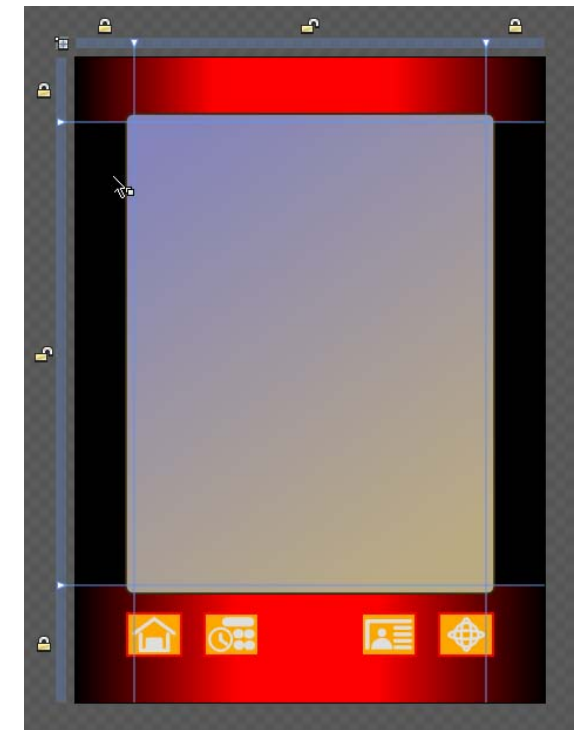




```
Dictionary1.xaml Home.g.cs Home.xaml.cs Home.xaml Calendar.xaml V
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/
  <DrawingBrush x:Key="ScreenBackground">
    <DrawingBrush.Drawing>
      <DrawingGroup>
        <GeometryDrawing Geometry="F1 M 250.4
          <GeometryDrawing.Pen>
            <Pen Thickness="2" LineJoin='
          </GeometryDrawing.Pen>
        <GeometryDrawing.Brush>
          <LinearGradientBrush StartPo:
            <GradientStop Color="#FF0
            <GradientStop Color="#FF0
          </LinearGradientBrush>
        </GeometryDrawing.Brush>
      </GeometryDrawing>
    </DrawingGroup>
  </DrawingBrush.Drawing>
</DrawingBrush>
<LinearGradientBrush x:Key="BorderTopBackground"
  <GradientStop Color="#FF000000" Offset="0"/>
  <GradientStop Color="#FF000000" Offset="1"/>
  <GradientStop Color="#FFFF0000" Offset="0.68'
  <GradientStop Color="#FFFF0000" Offset="0.29'
</LinearGradientBrush>
<SolidColorBrush x:Key="BorderRightBackground" Co
<SolidColorBrush x:Key="BorderLeftBackground" Co
<LinearGradientBrush x:Key="BorderBottomBackgrou
```





Laurent Bugnion  
Senior Software Engineer  
Siemens Building Technologies

- 12 years in software development
- 6 years experience with .NET (ASP.NET, C#, WinForms)
- 1.5 years working with WPF
- Microsoft MVP for ASP.NET
- What's GalaSoft?



## Traditional roles in UI development

- Graphics designer / UX designer
  - Deliverables:
    - Usability concepts
    - Styleguides
    - Screens (Photoshop, Powerpoint, etc...)
- UI developer
  - Does his (limited) best to translate the Graphics designer's wishes in a usable application with the chosen technology



## Enters XAML

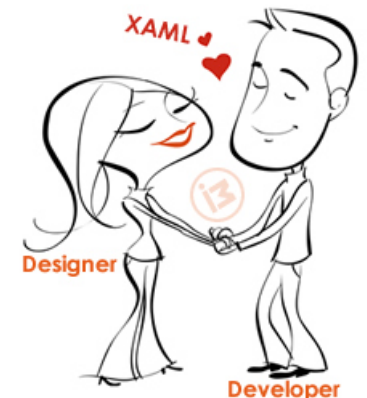
*eXtensible Application Markup Language (XAML)*

- Declarative markup language for UI design  
*(but not only)*
- A very large subset of the WPF API can be leveraged using entirely XAML
- Enables compelling tooling scenarios
- Separation between XAML front-end and code-behind  
*(Similar to ASP.NET)*
- Demo



## New roles thanks to XAML (and to the tools)

- Graphics designer as part of development team!
  - Deliverables:
    - Assets (XAML code)
- WPF Integrator
  - Enables a seamless-er workflow
  - Translates from designer-ish to developer-ish (and back)
  - Interface XAML – C#
  - « Devigner », ideally a senior developer with a taste for design





## Major tools used in WPF development

- Expression Design
- Expression Blend
- Visual Studio

### Historically

Current versions:

- Visual Studio 2005
- Expression Blend 1
- .NET 3.0

**Soon** (*27th of February 2008, now available as CTP*)

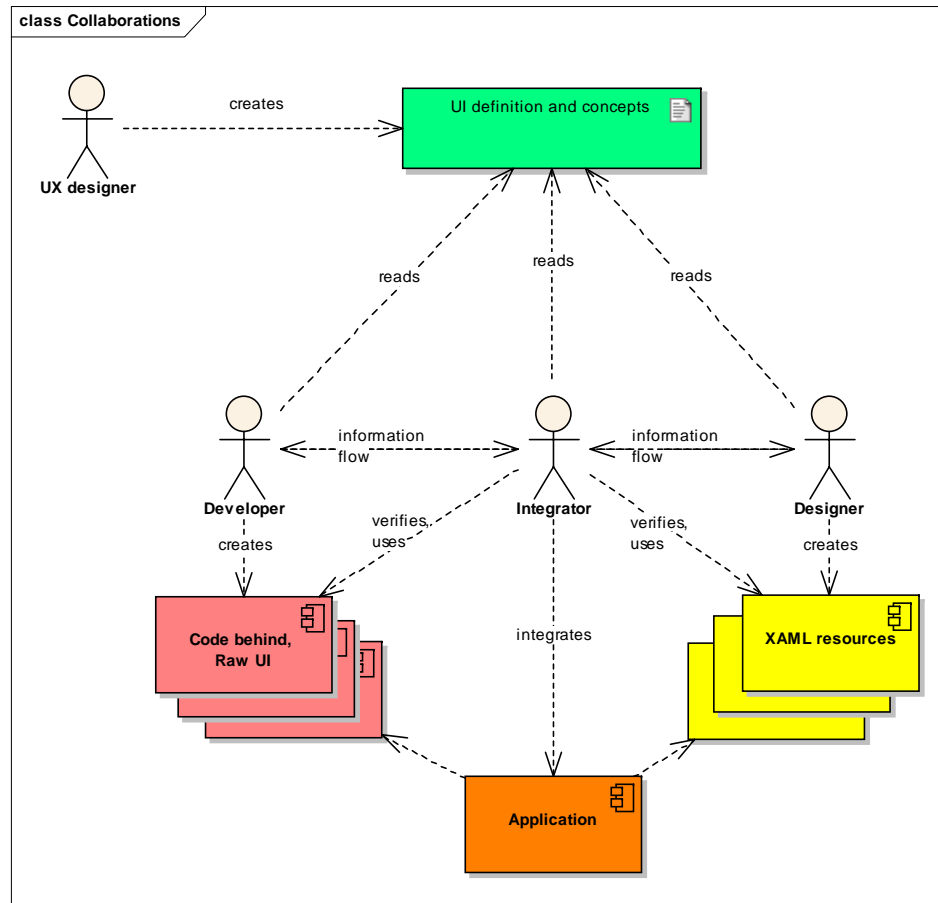
Visual Studio 2008 « Orcas »

Expression Blend 2

.NET 3.5



## Summary of roles and deliverables





## Differences towards « classical » workflow

### *UX specialist:*

The UI must not be specified until the last detail (pixels, shades, etc...). At a later stage of the development, the application's UI code (XAML) can be reworked and refined to reach the desired quality.

### *Graphics designer:*

The designer does not work on mockups and screens anymore, but on actual (XAML) code. He relies on new tools able to compile and run the application. He is a member of the development team!



## Differences towards « classical » workflow (cont.)

### *UI developer:*

The developer does not own the UI anymore. He merely creates a raw UI, which will be “filled” later by the integrator.

### *WPF integrator:*

This is a new role



## Demo



```
Dictionary1.xaml | Home.g.cs | Home.xaml.cs | Home.xaml | Calendar.xaml | V
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml"
  xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml">
  <DrawingBrush x:Key="ScreenBackground">
    <DrawingBrush.Drawing>
      <DrawingGroup>
        <GeometryDrawing Geometry="F1 M 250.4
          <GeometryDrawing.Pen>
            <Pen Thickness="2" LineJoin='
          </GeometryDrawing.Pen>
          <GeometryDrawing.Brush>
            <LinearGradientBrush StartPo:
              <GradientStop Color="#FF!
              <GradientStop Color="#FF!
            </LinearGradientBrush>
          </GeometryDrawing.Brush>
        </GeometryDrawing>
      </GeometryDrawing>
    </DrawingGroup>
  </DrawingBrush.Drawing>
</DrawingBrush>
<LinearGradientBrush x:Key="BorderTopBackground"
  <GradientStop Color="#FF000000" Offset="0"/>
<GradientStop Color="#FF000000" Offset="1"/>
<GradientStop Color="#FFFF0000" Offset="0.68'
<GradientStop Color="#FFFF0000" Offset="0.29'
</LinearGradientBrush>
<SolidColorBrush x:Key="BorderRightBackground" Co
<SolidColorBrush x:Key="BorderLeftBackground" Co
<LinearGradientBrush x:Key="BorderBottomBackgrou
```

