

PV239 Windows 10

Cvičení 1

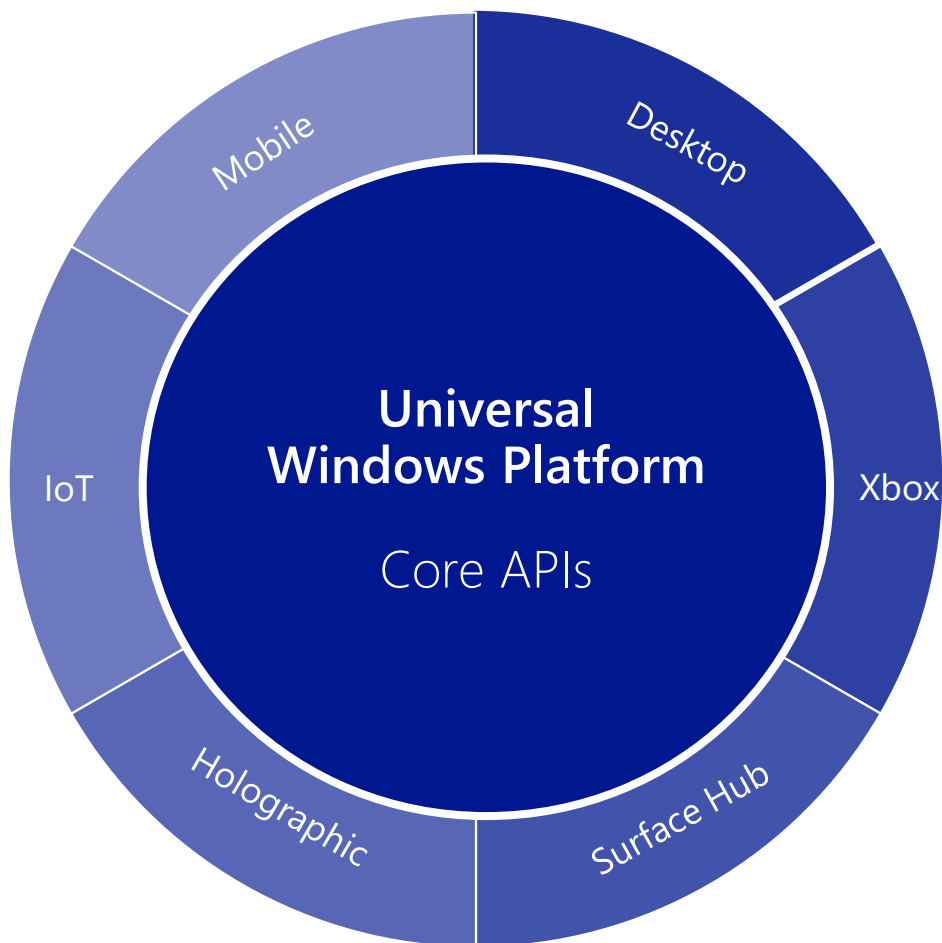
Mgr. David Gešvindr

MVP: Azure | MCSD: Windows Store | MCSE: Data Platform | MCT | MSP

gesvindr@mail.muni.cz

 @gesvindr

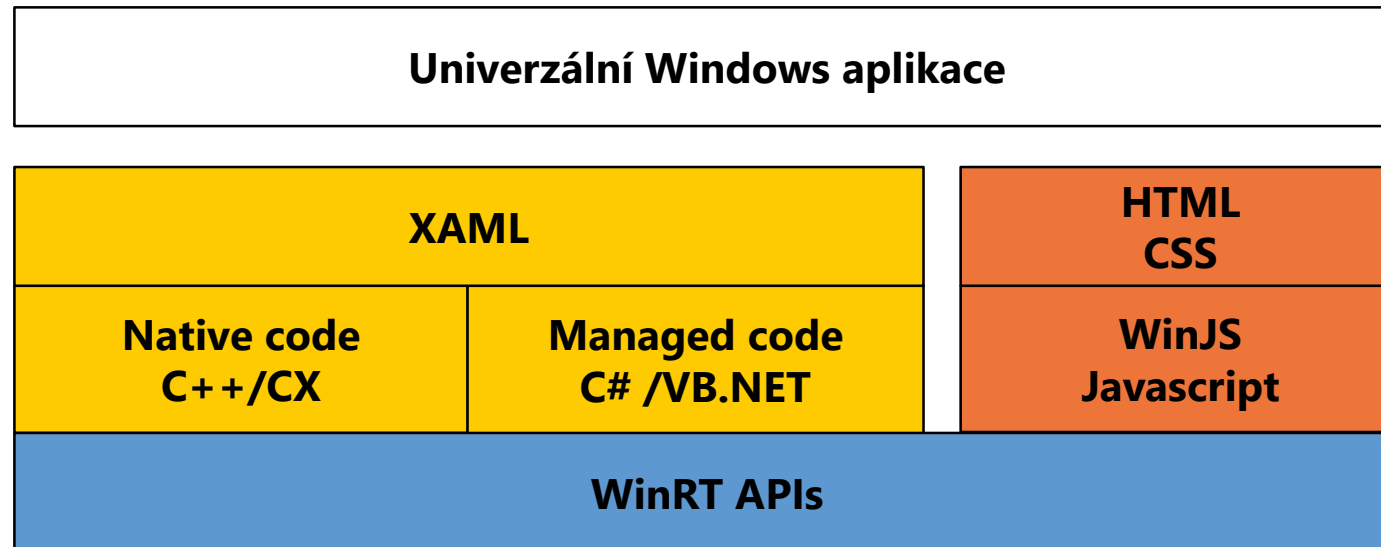
Universal Windows Platform



- Jednotná API na různých zařízeních
- Jeden appx balíček s aplikací
- Podpora adaptace na různá zařízení je součástí platformy

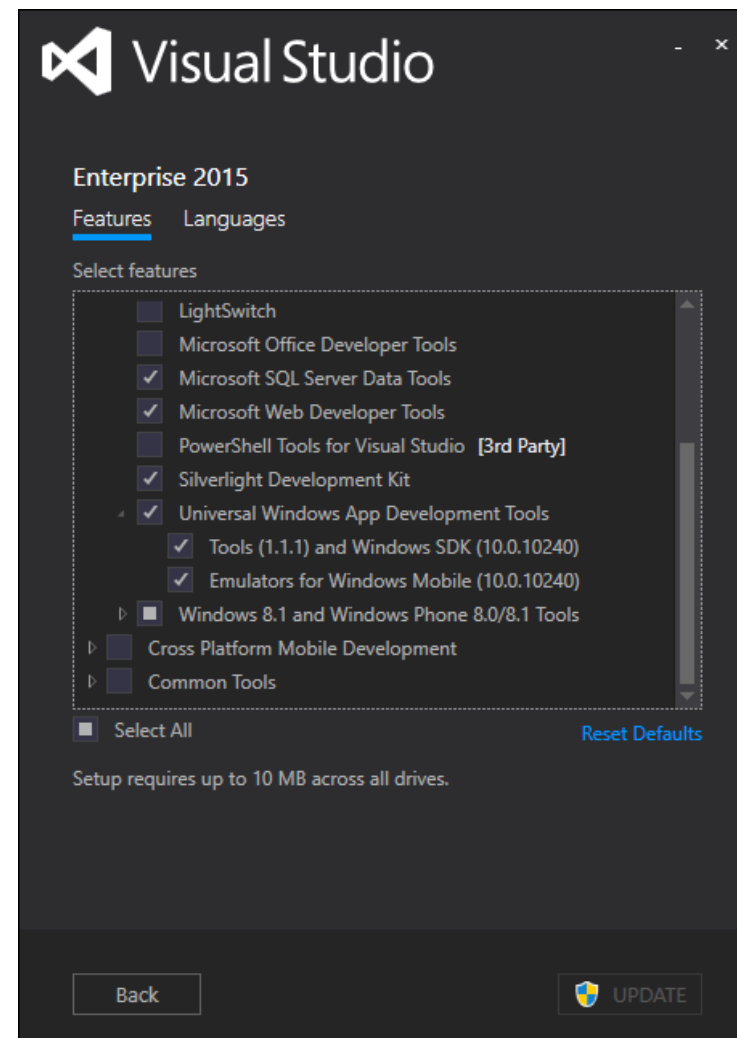
Jazyky pro tvorbu aplikací

- Runtime i dostupné jazyky jsou stejné jako u Windows 8.1 a Windows Phone 8.1
 - Jednoduchá migrace aplikací na novou platformu



Nástroje na tvorbu aplikací

- Visual Studio 2015
- Pro vývoj aplikací je třeba **Windows 10**
 - Zdarma edice **Community**
 - Placená edice **Professional** a **Enterprise**
 - Šablona: **Windows / Windows Universal / Blank App (Windows Universal)**
 - ◆ Zveřejnění dodatečných šablon není aktuálně v plánu

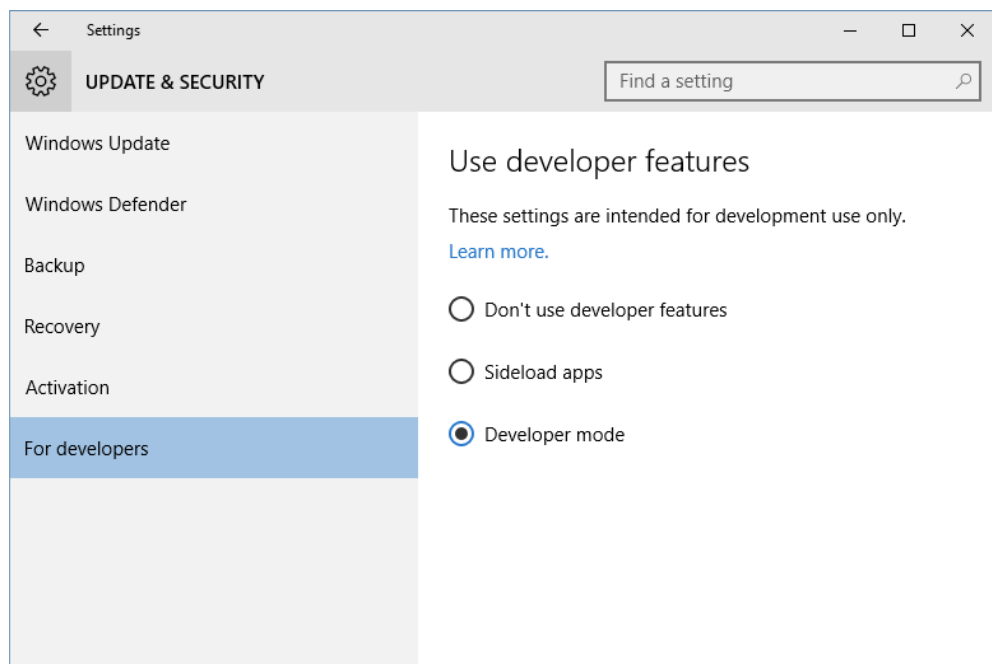


Ukázkové aplikace jako zdroj informací

- Microsoft již neposkytuje ve Visual Studiu nachystané šablony aplikací
 - Napáchaly víc škody, než užitku
- Místo toho zveřejnil na GitHubu celou sadu ukázkových aplikací pro Windows 10:
- <https://github.com/Microsoft/Windows-universal-samples>

Nasazení aplikace při vývoji

- Pro nasazení na počítač vývojáře je nutné systém odemknout
 - **Podstatně zjednodušeno**
 - **Neomezený** počet aplikací na Windows
 - Až 20 na telefonu

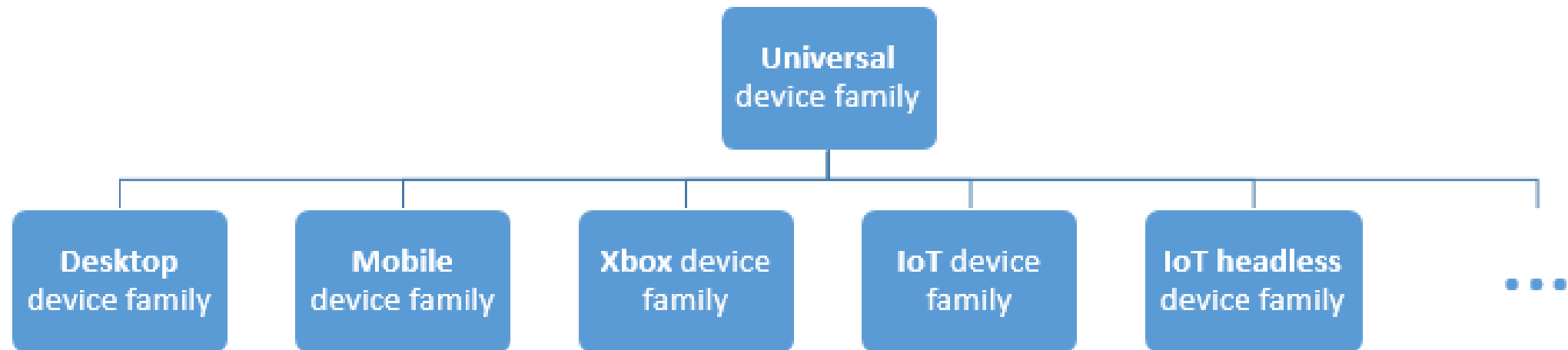


Aktualizace stávajících aplikací

- V tuto chvíli **není možné** ve Visual Studiu realizovat automatický upgrade
- Vytvořit nový projekt univerzální aplikace a postupně překopírovat stávající aplikaci
- Nejčastější problémy bránící spuštění:
 - Odebrání systémové resource (zejména Windows Phone)
 - Drobná změna API
- Nutné úpravy v UI aplikace:
 - Přidání adaptability na různá zařízení + úprava dle nových UI guidelines
- Aplikace na Windows Phone v Silverlightu vyžadují přepsat
 - Jak změna API, tak změna veliké části ovládacích prvků UI

Přizpůsobení aplikace zařízení

- Aplikace by se **neměla** cílit na telefon/tablet/počítač
- Aplikace by měla detekovat přítomnost specifických API a podle toho se přizpůsobit
- Přítomnost API je dána cílenou **device family**



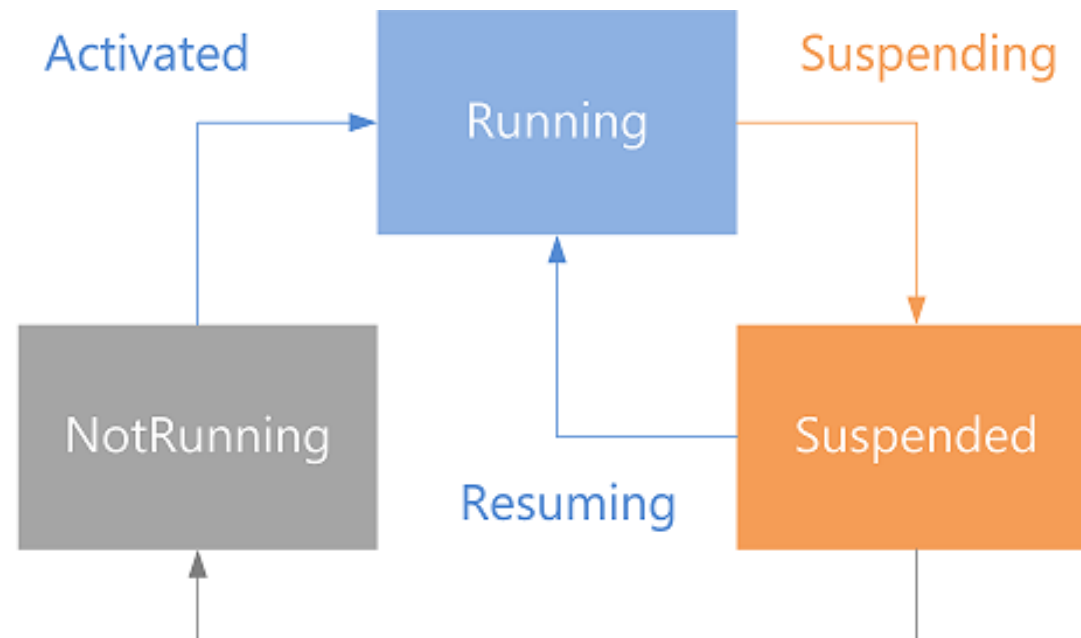
Jak psát aplikace napříč více typy zařízení

- Cílíte aplikaci na **Universal device family**
- Další device family přidáte jako **Extension SDK**
- Při použití API z Extension SDK musíte ověřovat jeho přítomnost na zařízení

```
if (Windows.Foundation.Metadata.ApiInformation.IsTypePresent("Windows.Phone.UI.Input.HardwareButtons"))  
{  
  
}
```

Životní cyklus aplikace

- Aplikace běží v sandboxu
- UWP adaptivně poskytuje prostředky pro běh aplikace v závislosti na aktuálním zařízení



Základní struktura aplikace

- Každá aplikace je tvořena **stránkami**
- Stránka je reprezentována třídou dědící z třídy *Page*
- Každé stránce odpovídá separátní XAML soubor a s ním související code-behind (C#)

Application (App.xaml)

Frame

Page

*MainPage (MainPage.xaml,
MainPage.xaml.cs)*

Seznámení se XAML

- **Extensible Application Markup Language** (XAML) je deklarativní jazyk který umí vyjádřit jakýkoliv strom CLR objektů
- XML element v XAML odpovídá .net třídě
- XML atributy jsou mapovány na vlastnosti objektu
- Zanoření elementů reprezentuje hierarchii objektů

- Každý XAML dokument má jen jeden ROOT element

Zápis XAML vs. C#

Zápis v XAML

```
<StackPanel>  
    <TextBlock Margin="20">Welcome to the World of XAML</TextBlock>  
    <Button Margin="10" HorizontalAlignment="Right">OK</Button>  
</StackPanel>
```

Identický zápis v C#

```
// Create the StackPanel  
StackPanel stackPanel = new StackPanel();  
this.Content = stackPanel;  
// Create the TextBlock  
TextBlock textBlock = new TextBlock();  
textBlock.Margin = new Thickness(20);  
textBlock.Text = "Welcome to the World of XAML";  
stackPanel.Children.Add(textBlock);  
// Create the Button  
Button button = new Button();  
button.Margin = new Thickness(10);  
button.Content = "OK";  
stackPanel.Children.Add(button);
```

Základní ovládací prvky

Button

Button

ToggleSwitch

WiFi



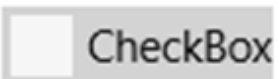
On

WiFi



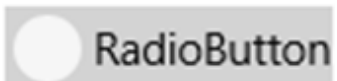
Off

CheckBox



CheckBox

RadioButton



RadioButton

Slider



ProgressBar

Creating photo album



Downloading files



Checking network requirements

TextBlock

Text

TextBox

Written text

Ovládací prvky pro rozložení obsahu

- Ovládací prvek **StackPanel**
 - Zobrazí zanořené ovládací prvky zarovnané do řady buď vertikálně nebo horizontálně
- Ovládací prvek **Grid**
 - Definuje řádky a sloupce do kterých se rozmísťují jednotlivé ovládací prvky
- Ovládací prvek **VariableSizedWrapGrid**
 - Ovládací prvky umístovány do řádku/sloupce a pokud je dosaženo max. počtu objektů, vznikne nový řádek/sloupec

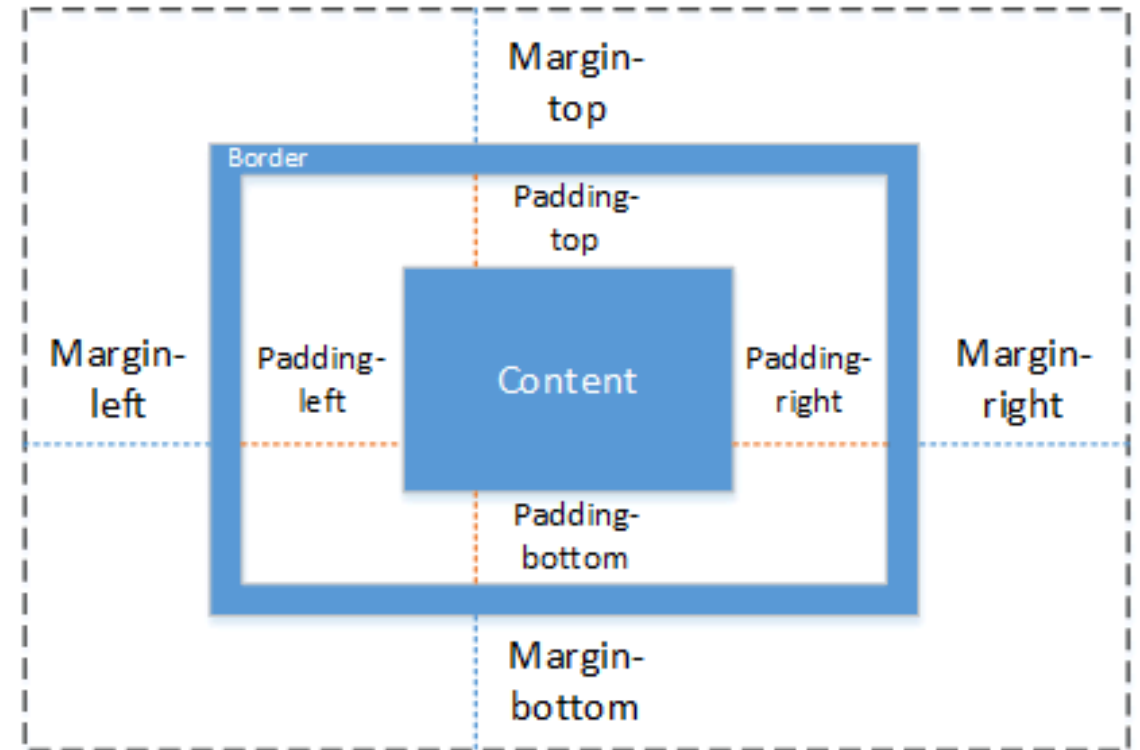
Ovládací prvky pro rozložení obsahu

- Ovládací prvek **RelativePanel**

- Umístění zanořených ovládacích prvků se popisuje relativně vůči sobě
- Používané vlastnosti:
 - ◆ Above
 - ◆ Below
 - ◆ LeftOf
 - ◆ RightOf
 - ◆ Align{Bottom|Left|Top|Right}With
 - ◆ Align{Bottom|Left|Top|Right}WithPanel

Hlavní vlastnosti

- **Width**
 - MinWidth, MaxWidth
- **Height**
 - MinHeight, MaxHeight
- **Margin**
- **Padding**
- **HorizontalAlignment, VerticalAlignment**
- HorizontalContentAlignment, VerticalContentAlignment



Implementace aplikační lišty

- S pomocí ovládacího prvku **AppBar**

```
<Page.BottomAppBar>  
  <AppBar x:Name="PageAppBar">  
    <!--Umístěte tlačítka-->  
  </AppBar>  
</Page.BottomAppBar>
```

- Prvky nutné uspořádat pomocí StackPanelu
- Podporované ovládací prvky:
 - **AppBarButton**
 - **AppBarToggleButton**
 - **AppBarSeparator**
 - Nastavují se atributy *Label* a *Icon*
 - Možnost využít Flyout

Implementace aplikační lišty

- Z ovládacího prvku **AppBar** dědí prvek **CommandBar**, který rozšiřuje jeho funkcionalitu
- Může obsahovat jen ovládací prvky implementující **ICommandBarElement**
- Automaticky řeší layout (netřeba StackPanel)
- Obsahuje
 - Primární seznam příkazů (vpravo)
 - Libovolný obsah (vlevo)
 - Sekundární seznam příkazů (ve vysunovacím menu)

