



# Erstellung einfacher Apps für Android, Windows Phone und iOS mittels Cross-Compiling

Jürgen Raab, BEd BSc

# Person

Jürgen Raab, BEd BSc

Lehrämter: M, GS, PC, Informatik, PTS

Lehrer an der PTS und  
NMS St. Georgen/Gusen seit 1997



Lehraufträge an der PH OÖ seit 2011 im Studiengang  
IKP (Informations- und Kommunikationspädagogik

Studium Software Engineering an der FH Hagenberg

# Wozu Apps entwickeln?

- Apps are everywhere!
- Informatikunterricht kann mehr bieten als den ECDL.
- Schwerpunkte in Fachbereichen der NMS.
- Vorbereitung auf den Beruf in der PTS.
- Benötigte Software ist kostenfrei verfügbar, jeder Schüler hat ein Smartphone!
- Kleines Programm große Wirkung!

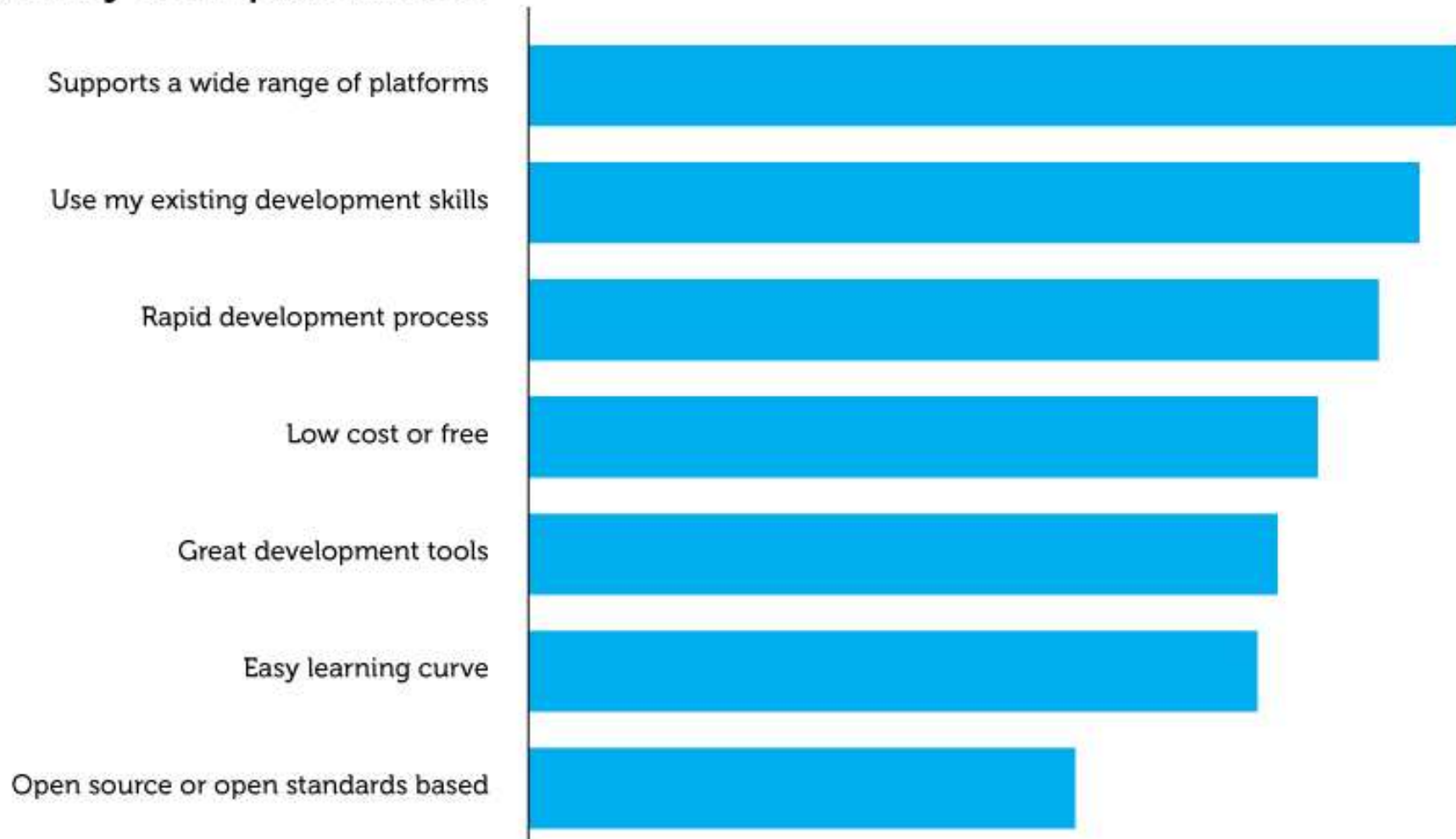
# Warum Cross-Compiling?

- Effiziente Entwicklung für alle Plattformen (Cross-Platform-Development)
- Nur eine Technologie muss vermittelt werden.
- Für alle Schüler am eigenen Gerät verwendbar
- Anknüpfungspunkte an bereits bekannte Inhalte oder Ausgangspunkt für eine vertiefende Beschäftigung mit App-Entwicklung

## DEVELOPER BENEFITS VALUED HIGHER THAN USER EXPERIENCE

% of developers using cross-platform tools by top 3 selection criteria (n=1,664)

### Reasons for selecting a primary cross-platform tool



Source: Cross-Platform Tools 2015 | vmob.me/CPT15 Copyright VisionMobile | All rights reserved

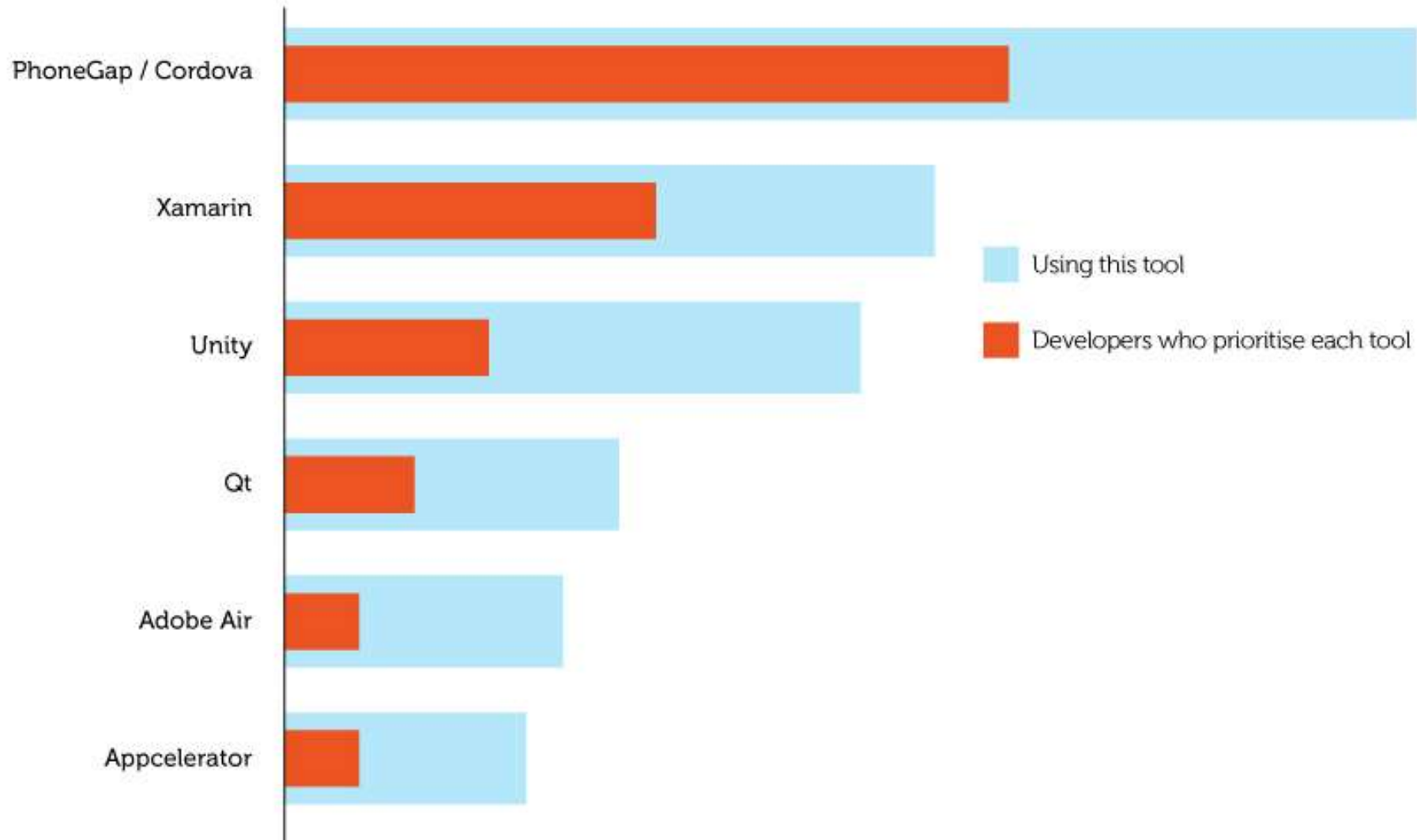


# Ansätze für die Umsetzung

- Reine Webanwendung mit responsive design
  - HTML5, CSS, Javascript -> Twitter Bootstrap
- Web-App
  - Webanwendung verpackt in Browser-Komponente -> PhoneGap/Cordova
- Cross-Compiler
  - JavaScript -> Appcelerator Titanium
  - C# -> Xamarin
  - ActionScript 3.0 -> Adobe Air

## THE TOP 3 CROSS PLATFORM TOOLS ACCOUNT FOR 70% OF PRIMARY USE

% of developers using cross-platform tools by primary tool and tools used (n=1,664)



Source: Cross-Platform Tools 2015 | [vmob.me/CPT15](http://vmob.me/CPT15) Copyright VisionMobile | All rights reserved

# Vor- und Nachteile PhoneGap/Cordova

- Vorteile:
  - Plug-Ins für Hardwarekomponenten
  - HTML, CSS und Javascript sind bekannte Technologien.
  - Für alle Plattformen verfügbar – open source
- Nachteile:
  - Performance
  - Qualität der UI oft mangelhaft
  - Größere Anwendungen mit plattformspezifischen Anforderungen oft schwierig – Eigene Plug-Ins?



# Vor- und Nachteile Titanium

## ➤ Vorteile:

- Einfach Entwicklung mit JavaScript und eigener API
- Gute Performance (native UI-Komponenten)
- Architektur (MVC)

## ➤ Nachteile:

- Zugriff auf Hardwarekomponenten manchmal schwierig
- Nicht für alle Plattformen verfügbar
- Plattform SDK's erforderlich

# Vor- und Nachteile Adobe Air

## ➤ Vorteile:

- Verfügbar für iOS, Android, WindowsPhone und Blackberry
- Verwendung von Flash-Movies möglich
- Einfache API mit professioneller IDE

## ➤ Nachteile:

- Eigene Programmiersprache
- Proprietäre IDE (Adobe Education)  
FlashDevelop als Alternative!
- Sinkende Bedeutung von Flash -> LTS PhoneGap?

# Allgemein

- Kein „write once, run everywhere“
- Look and Feel ist immer anders/schlechter als bei nativer App-Entwicklung
- Zugriff auf Hardwarekomponenten umständlicher als bei nativer Programmierung
- Betriebssysteme entwickeln sich sehr rasch

# Apps mit ActionScript 3.0

- IDE FlashBuilder 4.7 (proprietär) oder FlashDevelop (open source)
- Anwendungsbeispiel LIFEtool



# ActionScript-Entwicklungsumgebung

- Implementierung mit Adobe FlashBuilder 4.7 oder FlashDevelop
- Framework Robotlegs 2.2.1
- Flex SDK



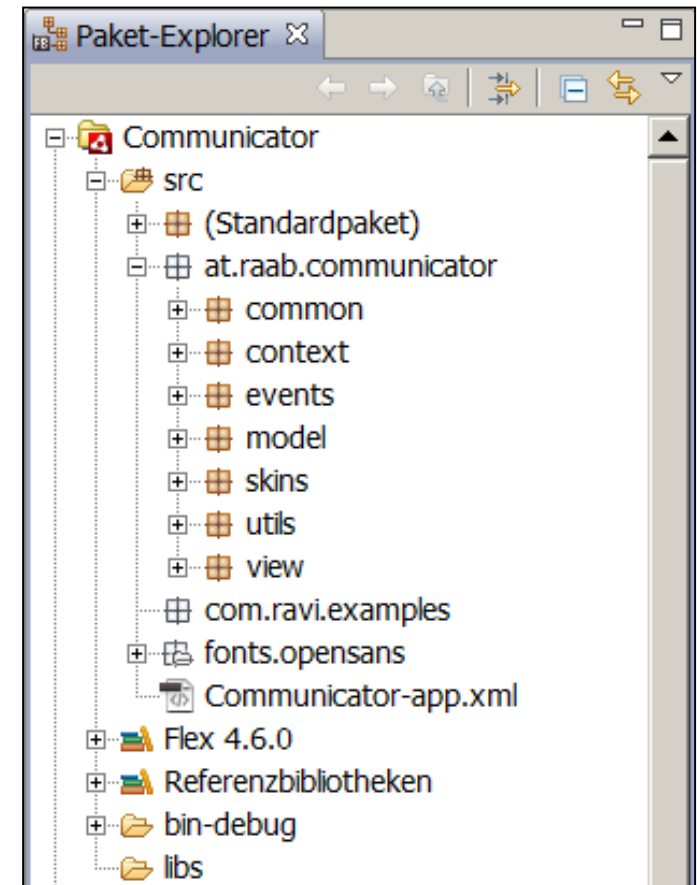
## Architektur

- MVCS – Architektur
- Events
- Dependency Injection



# Adobe FlashBuilder 4.7 Premium

- Ähnlich zu Eclipse
- Views mit MXML  
(Macromedia XML)
- ActionScript 3.0
- Adobe Flash Professional CS6



# Fragen?

