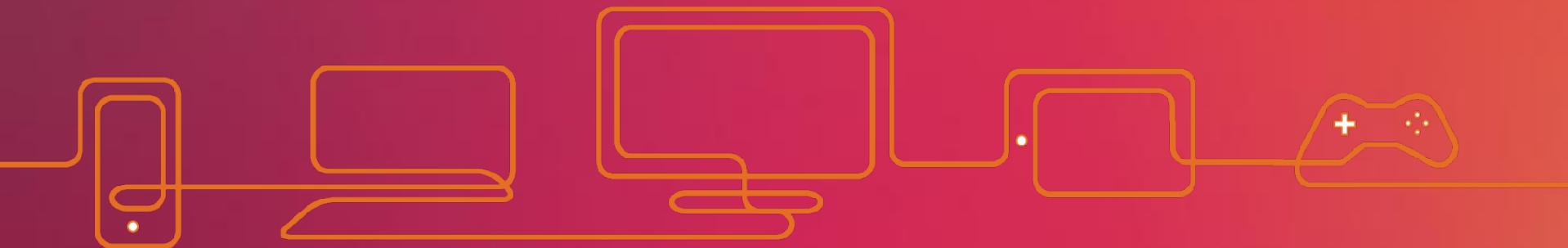




# First Contact

## Selling Haxe to Clients and Developers



Dominic De Lorenzo & David Peek

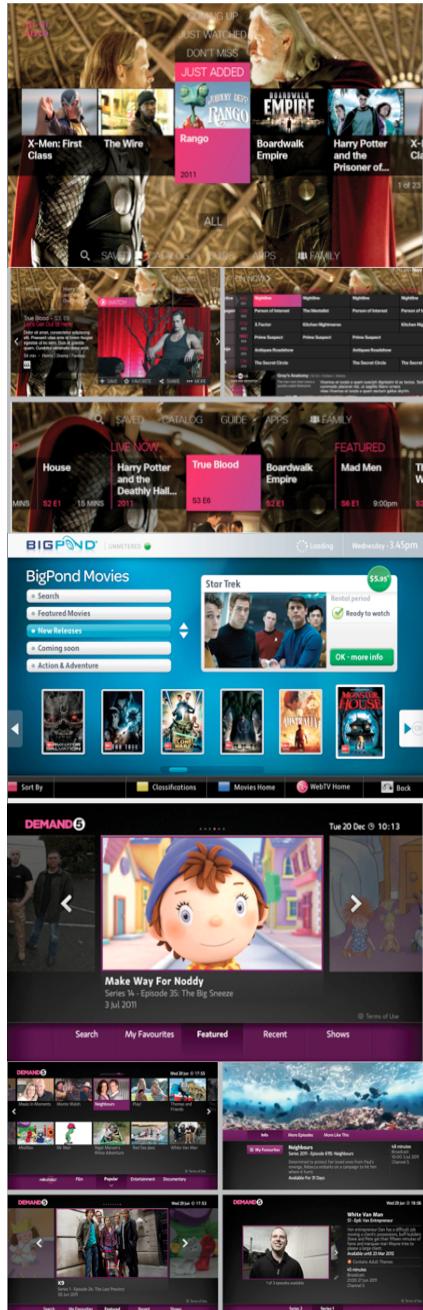
# Quick Introduction

- We are @DavidPeek & @misprintt
- We are from Sydney, Australia
- We both have many years experience building highly interactive web applications for all kinds of connected devices

# Quick Introduction

- We work at  **massive**

- TV Everywhere applications
- Sydney, London, New York
- Develop multi device user interfaces
- Commercial UI Framework written in Haxe



# Quick Introduction

- We **love** Haxe
  - 15+ Haxe developers
  - All UI development written in Haxe
  - Company wide commitment to Haxe
- We invest back into open source
  - MassiveUnit
  - MassiveCover
  - RobotHaxe
  - Hamcrest

# Quick Introduction

- We sell in Haxe to many of our clients
- It isn't always easy!

CISCO  
comcast.  
Panasonic

BBC

virgin atlantic  
Emirates  
IBERIA  
5  
BT

Tabcorp

FOXTEL

QANTAS

ABC

T

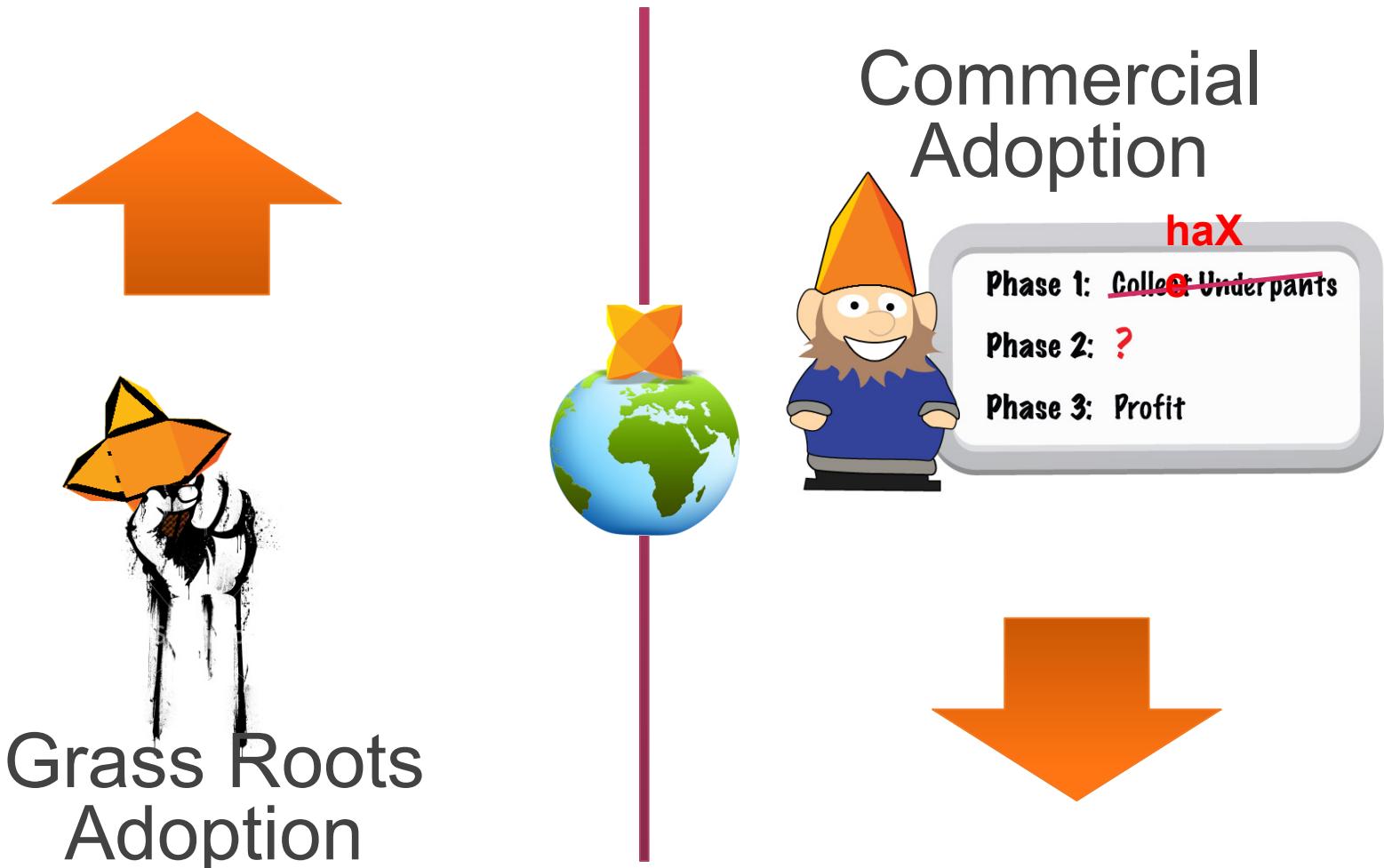
SKY

# We all have one common goal

To increase the broader adoption of Haxe

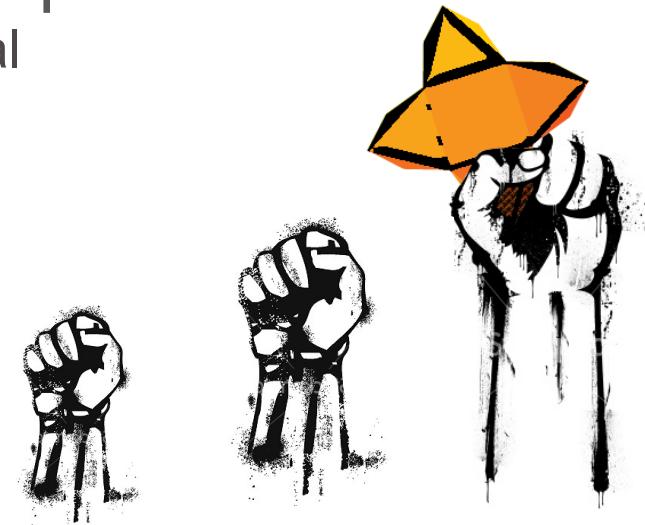


# We have **two** ways to get there



# Grass Roots Adoption

- The community is doing a great job
- Targeted communication
  - Different audiences care about different value propositions
  - HaxeJS, HaxeNME, HaxeNode
- Image makeover
  - Website, logo, mission statement
- Plants the seed for business adoption
  - Puts Haxe in peoples minds as a potential solution to problems (important first step)



# Commercial Adoption

- Adopting any new technology is **risk**
- Adopting any new technology takes time.
- There are multiple audiences to consider
- The final decision is rarely just technical
- The wrong choice can be **very expensive**
- Every business is different



# Know your audience

There are different audiences to consider





# Technical Expert

James the 'Neckbeard'

- Identify their credibility checklist
- Try and compare features and strengths to familiar technologies
- Steer clear of advanced features unless you have enough time

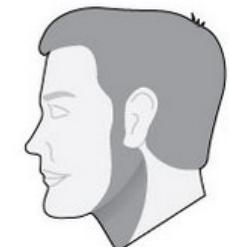
## It's a beard ... on your neck

Differing from the standard beard, a neckbeard is characterized by hair growth concentrated below the neckline.

BEARD



NECK BEARD



TRIBUNE GRAPHIC by Keith Claxton

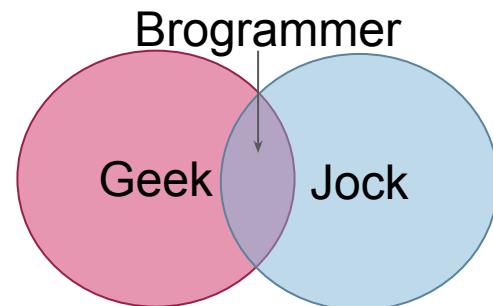


# Resident Developer

Jimmie the 'Brogrammer'

Brogrammer primer: [http://www.youtube.com/watch?v=Qi\\_AAqi0RZM](http://www.youtube.com/watch?v=Qi_AAqi0RZM)

- Its very easy to alienate developers up front
- Don't just highlight the limitations of their native platform
- Careful when talking outside their comfort zone and
- Focus on the day-to-day coding benefits





# Project Manager

Andrew the 'Down Trodden'

- Most likely to defer decision to other stakeholders
- Focus on delivery and predictability
- Focus on specific project challenges and demonstrate it has been done before





# Business Stakeholder

Ron the 'VP of Strategic Strategy'

- Focus on the big picture benefit to the business - ROI
- Highlight strategic benefits
- Demonstrate how Haxe gives the business options and flexibility
- Sprinkle with buzz words like HTML5, scalable, cloud, social, and synergy



# Technology is evaluated by four priorities

PERCEP  
UAL

How is it  
presented?



STRATE  
GIC

Where can it take  
us?



PRACTIC  
AL

Does it fit our  
processes?



TECHNICO  
AL

Does it solve our  
problem?



# Technology is evaluated by four priorities

PERCEP  
UAL

How is it  
presented?



STRATE  
GIC

Where can it take  
us?



PRACTIC  
AL

Does it fit our  
processes?



TECHNICO  
AL

Does it solve our  
problem?



# Technologies are like brands

- First impressions count
- Evaluated on consistency, tone, design, communication
- Clients will have preconceptions based on
  - limited technical knowledge,
  - web presence, social media, word of mouth.



# It's pronounced “hex” (with a capital X)

- Names are powerful things
- Developers know the “X” makes it run faster
- “Normals” think haXe looks gimmicky
- “Proper” languages are proper nouns
  - Ruby, Python, Perl, Java, Javascript
- At the very least, be consistent



# Introducing Haxe for the first time

Select relevant facts for audience

- Designed for cross platform development
- Compiles to JavaScript, Flash, C++ and more
- Open source
- Strictly typed
- Looks like JavaScript/ActionScript/Java

```
class HelloWorld
{
    static function main()
    {
        return new HelloWorld();
    }
    public function new()
    {
        trace("Hello World!");
    }
}
```

<http://www.haxe.org>



Concise Factual statement

Simple code snippet

Brand recall

Direct audience to potential next steps

# Technology is evaluated by four priorities

PERCEP  
UAL

How is it  
presented?



STRATE  
GIC

Where can it take  
us?



PRACTIC  
AL

Does it fit our  
processes?



TECHNICO  
AL

Does it solve our  
problem?

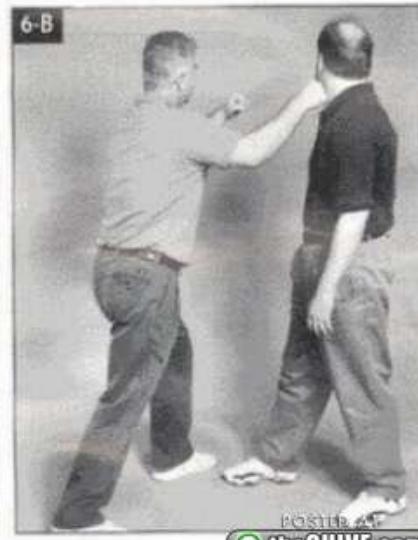
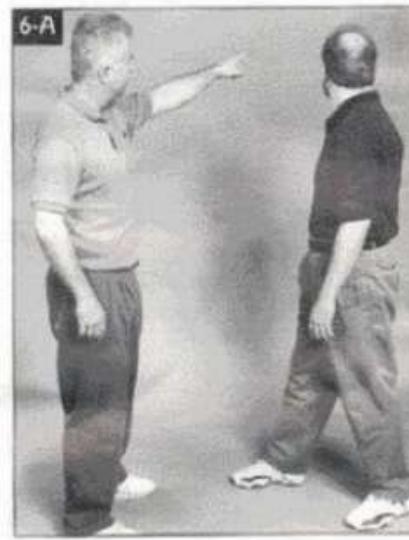


# Strategic

- Technology impacts a business's ability to innovate, evolve and grow.
- What opportunities can Haxe create?
- How can it provide an edge over competition using other solutions?

## TIPS-N-TRICKS

When they aren't looking...  
Punch 'em in the neck.



# The strategic advantage of Haxe

- Platform fragmentation is expensive
- Focus on Haxe's core strength to address the challenge
- Don't pretend Haxe is the only cross platform solution
- Outline how Haxe is different (and better)
- Demonstrate on-going strategic value of Haxe to the business

# Selling in the Have Advantage™

Big picture value to business

## Benefits for Business

- Reduce overall in-life cost of ownership
- Reduce time and effort (target both HTML5 and native)
- Avoid duplicated development streams per device
- Implement features and enhancements once
- Keep your options open (avoid the risk of investing in just one platform)



Icons are like a thousand buzz words

Reflections are shiny

Strategic benefits

# Technology is evaluated by four priorities



How is it presented?



Where can it take us?



Does it fit our processes?



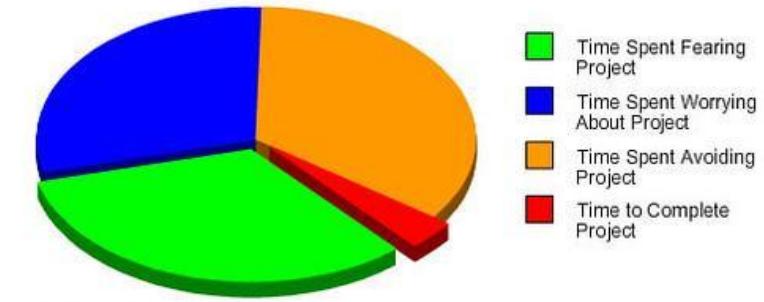
Does it solve our problem?



# Winning over Project Managers

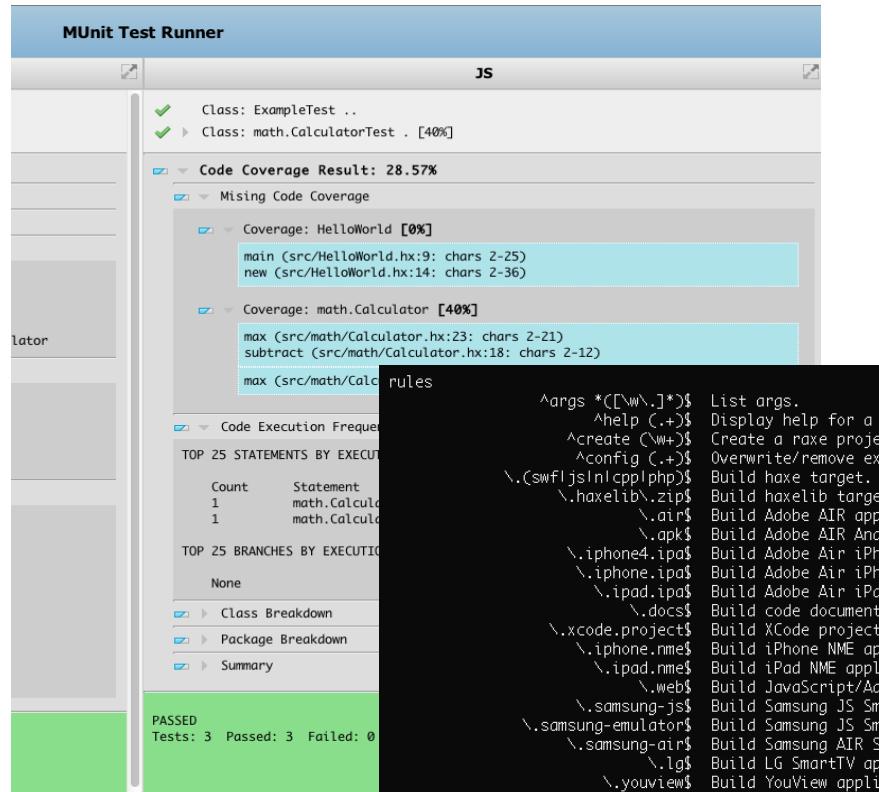
- Technology is a high risk decision
- Immediate impact
  - efficiency, quality, delivery
- Long term impact
  - profitability, innovation
- Focus of delivery processes
- Technology is a platform

**Project Accomplishment Schedule**



GraphJam.com

# Practical – Tooling



```
raxe[ task]
raxe[ rule match]

tasks
  default  The default task.
  init    Create ${project.build} and ${project.temp} directories.
  clean   Remove ${project.build} directory.
  clobber Remove ${project.temp} directory.
  help    Display help.
  tasks   List tasks.
  rules   List rules.
  version Display version.
  config  Open the raxe configuration module.
  display Print raxe completions target
  generate Build.hx
  build   Generate Build.hx
  server  Launch development server.
  refresh Refresh server.
  browse  Invoked on page refresh in index.n
  build.all Build all targets (web, air, apk, nme, ios, docs)
  munit_build Generates munit settings for project and temp test.hxml
  config.local GenerateDocsClass
  build.docs build docs
  configure.configure
  installer installer

  args *{([w+.]*)$ List args.
  ^help (+)$  Display help for a task/rule.
  ^create ([w+])$ Create a raxe project.
  ^config ([.]+)$ Overwrite/remove existing ConfigModule.
  \.(swf|fljs|ln|cpp|php)$ Build haxe target.
  \.haxelib\|.zip$ Build haxelib target.
  \.air$ Build Adobe AIR application.
  \.apk$ Build Adobe AIR Android application.
  \.iphone4.ipa$ Build Adobe Air iPhone4 packaged application.
  \.iphone.ipa$ Build Adobe Air iPhone packaged application.
  \.ipad.ipa$ Build Adobe Air iPad packaged application.
  \.docs$ Build code documentation.
  \.xcode.project$ Build XCode project.
  \.iphone.nme$ Build iPhone NME application.
  \.ipad.nme$ Build iPad NME application.
  \.web$ Build JavaScript/Adobe Flash web application.
  \.samsung-js$ Build Samsung JS SmartTV application.
  \.samsung-emulator$ Build Samsung JS SmartTV application.
  \.samsung-air$ Build Samsung AIR SmartTV application.
  \.lg$ Build LG SmartTV application.
  \.youview$ Build YouView application.
  \.xoom$ Build Xoom application.
  ^deploy (+)$ Build and deploy a target.
  ^build (+)$ Build a target.
  ^debug (+)$ Debug a target.
  ^launch (+)$ Launch a target.
  ^amui ([w+])$ [Deprecated - use 'create mui'] Create a MassiveUI project.
  ^create mui ([w+])$ Create a MassiveUI project.
  ^create class (for ?){([.w+]*)$ Create class, or class for a test class.
  ^create test (for ?){([.w+]*)$ Create a test class, or test class for class.
  ^test([.]*)$ Wrapper for 'munit test'. Run 'haxelib run munit help test' for details.
  ^munit ([.]*)$ Wrapper for 'munit'
  asset{ ([w+ ?)*?}([d+.]+)$ Generate asset libraries by priority of targeted assets. e.g. raxe ipad tablet large.
  ^release haxelib\|.zip ([d+.]+)$ Build a versioned haxelib archive.
dominic MassiveUI$
```

# Technology is evaluated by four priorities

PERCEP  
UAL

How is it  
presented?



STRATE  
GIC

Where can it take  
us?



PRACTIC  
AL

Does it fit our  
processes?



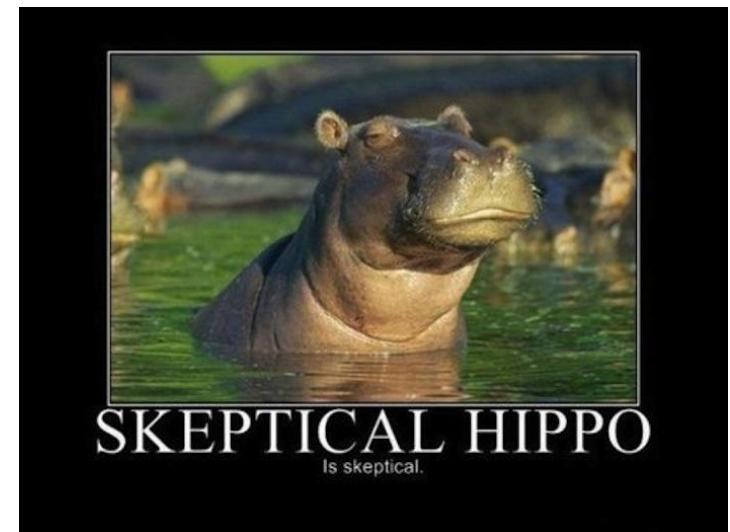
TECHNICO  
AL

Does it solve our  
problem?



# Developers, developers, developers

- Tough audience
  - Loyal to what they know
  - Skeptical of magic
- All the hard questions
  - Support – documentation, community, libraries
  - Tooling – IDE, build, deploy
  - Performance
- Use familiar comparisons if possible
- Identify weaknesses
  - Set tone for discussion



# Technical

- Address perceived weaknesses
  - Performance, documentation, support
- Acknowledge actual weaknesses
  - Tooling, developer base, non native experience
- Focus on strength of language and compiler
  - Functional/OPP, Macros, Neko,
- Embrace and extend their platform
  - Externs, haxelib, native extensions



# Technical

Rich language benefits

## Expressiveness

- Modern language
  - Metadata, RTTI, enums, generics
- Compact syntax, support for functional/OOP styles
- Powerful type system
- Compiler and macros are powerful developer tools
  - DSLs, code generation, e.g. HxSL
  - Neko allows developers to automate their workflow
- HaxeLibs are tools, not just libraries
- You're never far from the platform
  - Externs, native extensions, \_\_magic\_\_

More than just a language

Haxe embraces native rather than competes with it

# FAQ

Frequently Annoying Questions



# Frequently Annoying Questions



- Haxe can also do all the HTML5 things
- Haxe enables a better language and workflow for writing HTML5
- Haxe can also provide alternatives where HTML5 isn't available

# Frequently Annoying Questions



- Haxe can help you reach more platforms for less effort
- Haxe can save money by implementing features and enhancements once
- Haxe can provide a more streamlined developer workflow

# Frequently Annoying Questions



- Without Haxe that is still two totally separate development streams.
- What about all the customers on other platforms?
- Haxe can keep you on top of the ever shifting device/platform landscape

# Frequently Annoying Questions



- Granted there is a slight overhead in generated code.
- The more important question is whether the end user can notice?
- Haxe is used by many game studios where performance is critical

# Frequently Annoying Questions



But my hand  
crafted code is  
much prettier than  
that generated  
code...

- Usually a sign a developer feels out of their depth
  - Make sure to compare source code with source code.
  - Optimal code isn't always pretty code
- 

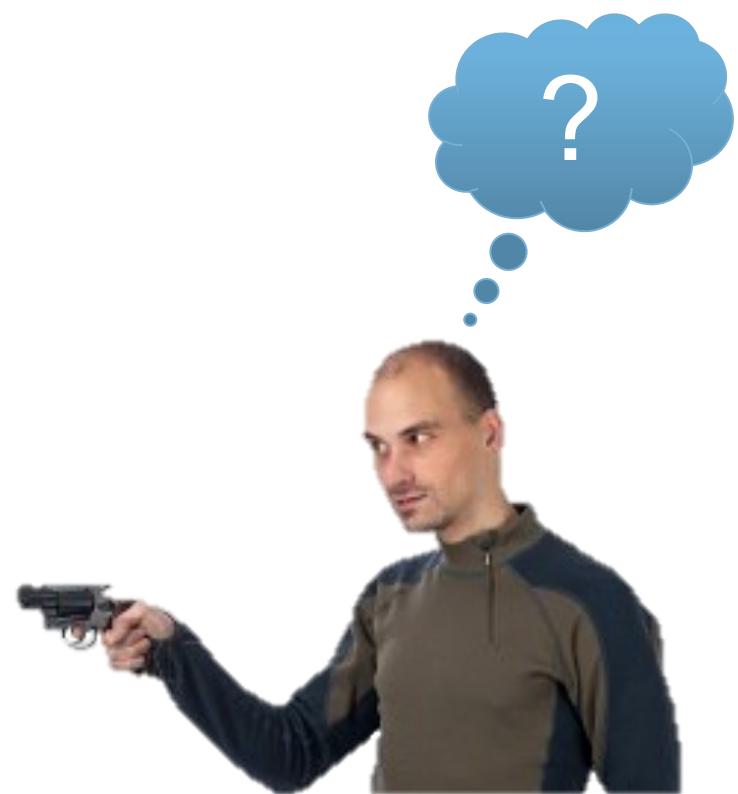
# Action points

For the Haxe Community

# Community Action Points

- Better visibility of existing libraries
  - Most popular
  - Comparisons/recommendations of existing libraries
  - Clearer indication of individual platform support
- Less individual libraries, more collaborative libraries
  - Many libraries aren't approachable (black box of features)
  - Less duplication of features
  - Better package management (smarter version dependencies, don't enforce license)
- Introduce coding best practice standards
- Consistent documentation
  - Separated versioned docs
  - Language Specification
- Consistent build process for multi-platform projects

# Any Questions?



# Thank you

- Follow us
  - @DavidPeek, @misprintt,
  - @MikeStead, @DeanBurge
- Work with us
  - [www.massiveinteractive.com/hiring](http://www.massiveinteractive.com/hiring)
- Fork us, baby
  - [www.github.com/massiveinteractive](http://www.github.com/massiveinteractive)
- Use us (our slides anyway)
  - <http://ui.massive.com.au/talks/>