

Building a community as a service



Unvanquished: Building a community as a service



Speaker:

Thomas “illwieckz” Debesse

- **Unvanquished** project head
- Contributors to various related open source projects

Unvanquished: Building a community as a service



Game:

Unvanquished

- Real Time Strategy game
- First Person Shooter point of view
- Free and open from code to data

Unvanquished: Building a community as a service



Speaker:

Thomas “illwieckz” Debesse

- SysAdmin
- Site ~~Reliability~~ Rehabilitation Engineer

Debian defines work methodology and builds mindset to think about problems

Unvanquished: Building a community as a service



Speaker:

Thomas “illwieckz” Debesse

A service is:

- **An act of being of assistance to someone**

Unvanquished: Building a community as a service



Game:

Unvanquished

- Powered by the Dæmon game engine,
→ a **grand-grand child of many forks**
including Quake 3

Unvanquished: Building a community as a service



A story of id Tech engines

A story of id Tech engines



- Wolfenstein 3D
- **DOOM** 1993 release
- **DOOM** 1999 opening

A story of id Tech engines



- Wolfenstein 3D
- DOOM 1993 release
- Quake 1996 release

- D&Q 1999 opening

A story of id Tech engines



- Wolfenstein 3D
- DOOM 1993 release
- Quake 1996 release
- **QII 1997 release**
- D&Q 1999 opening
- QII 2001 opening

A story of id Tech engines



Quake I & II derivatives

- **Kingpin, Soldier of Fortune, Half Life...**

A story of id Tech engines



id Software business

- **Selling games and code snapshots**

A story of id Tech engines



Quake I & II derivatives

- Kingpin, Soldier of Fortune, Half Life...
- Nexuiz, Xonotic, Warsow, Alien Arena, UFO: Alien Invasion, DDay: Normandy, Quetoo...

A story of id Tech engines



- Wolfenstein 3D
- DOOM 1993 release
- Quake 1996 release
- QII 1997 release
- QIII 1999 release
- D&Q 1999 opening
- QII 2001 opening
- QIII 2005 opening

A story of id Tech engines



Quake III derivatives

- **Return to Castle Wolfenstein, Wolf:ET, Elite Forces, Soldier of Fortune, Medal of Honor, 007, Resident Evil, Call of Duty...**

A story of id Tech engines



id Software opening

- Releases as **open source when obsolete** from an economic point of view

A story of id Tech engines



Quake III derivatives

- Return to Castle Wolfenstein, Wolf:ET, Elite Forces, Soldier of Fortune, Medal of Honor, 007, Resident Evil, Call of Duty...
- Tremulous, Unvanquished, Smokin' Guns, World of Padman, OpenArena...

A story of id Tech engines



- Wolfenstein 3D
- DOOM 1993 release
- Quake 1996 release
- QII 1997 release
- QIII 1999 release
- D&Q 1999 opening
- QII 2001 opening
- **DIII 2004 release**
- **QIII 2005 opening**
- QIV 2005 release
- DIII 2011 opening

A story of id Tech engines



Doom III derivatives

- Quake 4, Enemy Territory: Quake Wars, Wolfenstein, Prey...
- The Dark Mod...

Unvanquished: Building a community as a service



id Tech ≠ game engine

redistributable and usable with the GPLed engine code. One must purchase a copy of *Quake* in order to receive the registered version of the game which includes more single-player episodes and the *deathmatch* maps. Based on the success of the first *Quake* game, and later published *Quake II* and *Quake III Arena*; *Quake 4* was released in October 2005, developed by **Raven Software** using the *Doom 3* engine.

en.wikipedia.org

Unvanquished: Building a community as a service



Code licenses:

- GPLv2
- GPLv2+
- GPLv3

A story of id Tech engines



- Wolfenstein 3D
- DOOM 1993 release
- Quake 1996 release
- QII 1997 release
- QIII 1999 release
- D&Q 1999 opening
- QII 2001 opening
- DIII 2004 release
- QIII 2005 opening
- QIV 2005 release
- **DIII 2011 opening**
- **id Tech 5, 6, 7 ???**

Unvanquished: Building a community as a service



Competitor: Epic and Unreal Engine

Competitor: Epic and Unreal engine



Game:

Unreal Tournament

- Direct competitor of Quake III Arena

Engine:

Unreal Engine

Competitor: Epic and Unreal engine



Moddability

- making modification possible
≠ **designing for modification**

Competitor: Epic and Unreal engine



Game:
Unreal Tournament

- Editor first

Engine:
Unreal Engine

Competitor: Epic and Unreal engine



gamedev.net ↓

T [_the_phantom_](#) ❤️ 11,263
May 27, 2003 04:41 PM

quote: *Original post by Mihail121*

And i'm pretty sure that thing like "Unreal Engine" doesn't exist. The engine used in ALL unreal games is "Unreal Warfare" optimized and changed for each new Unreal game.

Pulled from the UDN page (Unreal Dev. Network)

"The Unreal Developer Network is a group of sites and services providing support and resources for licensees employing the latest builds of Epic Games' Unreal engine (builds 600+)."

So, yes it IS called the Unreal Engine, NOT "Unreal Warfare".

The engine is constantly under developement (thus the build numbers) and is only feature frozen on game release (if you lience the engine from Epic you can get updates to the engine all the time).

your **BEST RESOURCE** for the **UNREAL ENGINE**

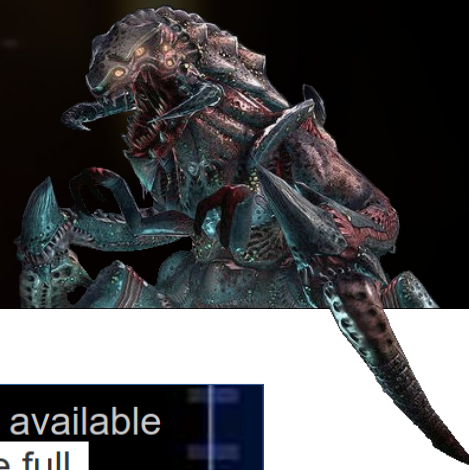
Welcome to UDN!

The **Unreal Developer Network** is a group of sites and services providing support and resources for licensees employing the latest builds of Epic Games' Unreal engine (builds 600+). Developers under non-disclosure agreements can [log in](#) to access all resources.

Important information as of October 5, 2002!

↑ [udn.epicgames.com](#)
(web archive, 2003)

Competitor: Epic and Unreal engine



The engine is currently supported on PC, PlayStation2 and Xbox. Source code is also available for any Epic-owned port including Macintosh, Linux and Dreamcast. Licensees receive full source code for our engine, tools and most recent game. Support is provided directly from the team that actually develops the engine and whose last two game titles sold over a million units and both won Game of the Year awards! We also have the experience of shipping a successful

Announcing the Unreal Developer Network!

UDN is a repository of knowledge, documentation and tutorials for the recent builds of the Unreal Engine.



I'm confused is it the Unreal II Engine, the UT2003 Engine, the Unreal Warfare Engine, the Unreal Championship Engine or the Unreal Tournament Engine?

It is neither. It is simply the "Unreal Engine". You'll sometimes hear it referred to as any one of these but as far as what licensees get there is really no distinction. During development licensees get continued access to all of Epic's technology updates and licensees are the ones to decide when they're going to stop using new updates and ship their game. For more

epicgames.com (web archive, 2003)

Competitor: Epic and Unreal engine



- Unreal Developer Network
- Reusable engine builds
- Reusable editor
- One engine

Competitor: Epic and Unreal engine



Community first

- Unreal Developer Network
- Reusable engine builds
- Reusable editor
- One engine

Unvanquished: Building a community as a service



Competitor: Valve and Source engine

Competitor: Valve and Source engine



Source code based on Quake I & II

- **Gold**

Competitor: Valve and Source engine



Source code based on Quake I & II

- GoldSrc
Half Life
- Source

Competitor: Valve and Source engine



Engine:
Source engine

- Engine **meant to be reusable**
→ based on id Tech code

Competitor: Valve and Source engine



Engine:
Source engine

- Engine meant to be reusable
- Editor and **tools designed to be reusable**

Competitor: Valve and Source engine



- Source engine builds
- Hammer editor and tools

Competitor: Valve and Source engine



- Source engine builds
- Hammer editor and tools
- Steam software distribution
→ solve file duplication and feature fragmentation

Competitor: Valve and Source engine



- Source engine builds
- Hammer editor and tools
- Steam software distribution
- Game ports

Competitor: Valve and Source engine



Community first

- Source engine builds
- Hammer editor and tools
- Steam software distribution
- Game ports

Unvanquished: Building a community as a service



On Open Source side of things : Godot

On Open Source side of things: Godot



- Free & Open Source
- Reusable builds

On Open Source side of things: Godot



- Free & Open Source
- Reusable builds
- Not a game
- **The engine for your game**

On Open Source side of things: Godot



Community first

- Free & Open Source
- Reusable builds
- Not a game
- The engine for your game

Unvanquished: Building a community as a service



**id Software engines, editors
and tools were templates**

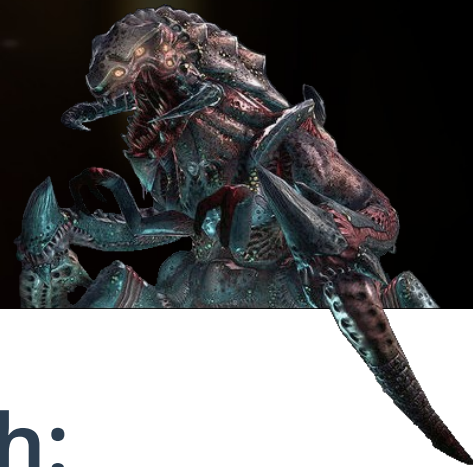
id Software engines, editors and tools were templates



To make a game on id Software tech:

- **fork the engine**
- build the engine
- maintain the engine

id Software engines, editors and tools were templates



To make a game on id Software tech:

- fork the engine
- build the engine
- maintain the engine

id Software sold **game templates**

Old Software engines, editors and tools were templates



Design induced mindset

- fork the engine
- build the engine
- maintain the engine

“It was always done this way”

Fixed Software engines, editors and tools were templates



Fixing design induced mindset

- requires an engine redesign

id Software engines, editors and tools were templates



Fixing code induced mindset

- requires an engine redesign
- requires **methodology shift**
- requires tool inducing good practices
→ like Git

Unvanquished: Building a community as a service



File formats meant to be forked

File formats meant to be forked



Example: **BSP** file format

- like a primitive tarball
 - **expect forks to have custom format**

File formats meant to be forked



Example: BSP file format

- like a primitive tarball
- **amount of sub files is not stored**

File formats meant to be forked



Example: BSP file format

- like a primitive tarball
- amount of sub files is not stored

you have to fork the format to add one file or change sub file format

File formats meant to be forked



Example: BSP file format

- like a primitive tarball
- amount of sub files is not stored

you have to fork the format to add one file

you have to fork engine and tools anyway

File formats meant to be forked



Example: SomaZ's Blender BSP plugin

- **no one want to fork it for every game**
- **every format would have to be supported**

File formats meant to be forked



id Software methodologies

- fit perfectly if targeting console cartridges

despite targeting PC market from the start

File formats meant to be forked



Example: BSP file format

- When adding features or fixing bugs
 - lose compatibility with existing assets
 - lose compatibility with existing tools

Unvanquished: Building a community as a service



Editors meant to be forked

Editors meant to be forked



- Quake level editor:
Quake ED

- Quake 3 level editor:
QERadiant

then
Q3Radiant

Editors meant to be forked



- Quake 3 level editor:
Q3Radiant
- Call of Duty level editor:
CoDRadiant
- Medal of Honor level editor:
MoHRadiant
- (...)

Editors meant to be forked



AARadiant, CoDRadiant, DarkRadiant, Embrace, ETXRadiant, GtkRadiant, HMF2Radiant, JK2Radiant, KpQ3Radiant, KpRadiant, MacRadiant, MoHRadiant, NetRadiant, NRC, ODRadiant, OWRadiant, QERadiant, Q3Radiant, QioRadiant, QtRadiant, Radiant, SDRadiant, SoFRadiant, ÜberRadiant, UFORadiant, VolatileRadiant, VRadiant, WolfRadiant, XreaLRadiant, ZeroRadiant (...)

Editors meant to be forked



AARadiant, CoDRadiant, DarkRadiant,
Embrace, ETXRadiant, GtkRadiant,
HMF2Radiant, JK2Radiant, Lepo3Radiant
KpRadiant, MacRadiant, MoHBRadiant
NetRadiant, NRC, QDRadiant, OWRRadiant,
QERadiant, Q3Radiant, QipRadiant, QtRadiant,
Radiant, SDRRadiant, SoFRadiant, ÜberRadiant,
UFORadiant, VolatileRadiant, VRadiant,
WolfRadiant, XreaLRadiant, ZeroRadiant (...)

bloodmap, etxmap2,
daemonmap, q3map2,
kpmmap2, vmapc,
ufo2map, xmap2 (...)

Editors meant to be forked



AARadiant, CoF, Radiant,
Embrace, ETX, Radiant,
HMF2, Radiant,
KpRadiant, M, Radiant,
NetRadiant, OWRRadiant,
QERadiant, Radiant, QtRadiant,
Radiant, Radiant, ÜberRadiant,
UFORadiant, VRadiant,
WolfRadiant, XreaLRadiant, ZeroRadiant (...)

bloo
daem
k
map2,
map2,
apc,
(...)

Editors meant to be forked



Outside of Radiant level editor

- no DarkBlender
- no NetBlender
- no ÜberBlender

Editors meant to be forked



Radiant level editor

- **doesn't expect you work on more than one game**

Editors meant to be forked



Radiant level editor

- doesn't expect you work on more than one game
- **requires your files to be stored at one hardcoded place**

Editors meant to be forked



Radiant level editor

- doesn't expect you work on more than one game
- requires your files to be stored at one hardcoded place
- **OK-ish for employees, not for indies**

Editors meant to be forked



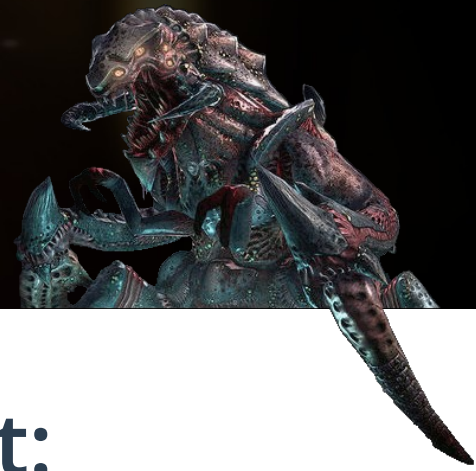
Radiant level editor

- at Quake Live time,

id Software forked GtkRadiant 1.4
while community used GtkRadiant 1.5

→ **functionality split**

Editors meant to be forked

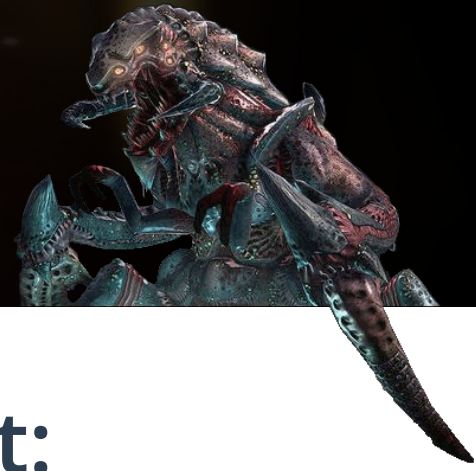


Some features added to NetRadiant:

- **open file from file manager** (2017)
- **load data from arbitrary paths** (2017)
- **change game without quitting** (2019)

to a software from before year 2000

Editors meant to be forked



Some features added to NetRadiant:

- open file from file manager (2017)
- load data from arbitrary paths (2017)
- change game without quitting (2019)
- **install following FHS standard (2018)**

Editors meant to be forked



Some Radiant forks in Debian

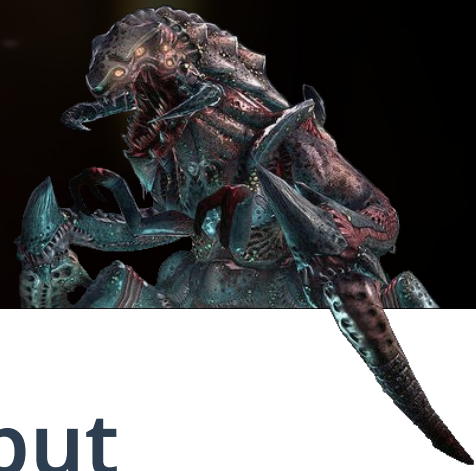
- **DarkRadiant**
NetRadiant fork
focused on The Dark Mod and Doom 3
- **UFORadiant**
DarkRadiant fork
UFO: Alien Invasion only

Unvanquished: Building a community as a service



On software development and child sacrifice

On software development and child sacrifice



id Software technology is efficient but

- design **assumes you fork** and never contribute back
- is **meant to be sold and forgotten**
- formats are **not meant to be extended without breaking compatibility**

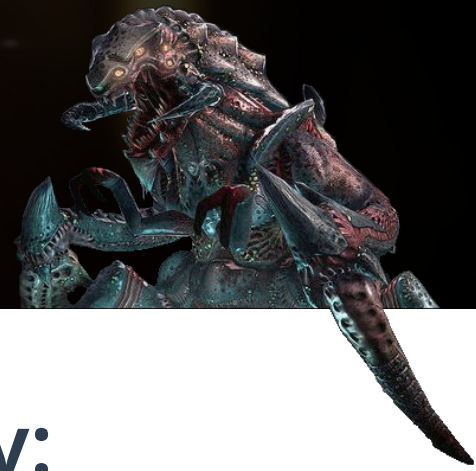
On software development and child sacrifice



id Tech engines are designed to
ship a game that will kill the previous one

- design is thought around that mindset
- **design induces that mindset again**

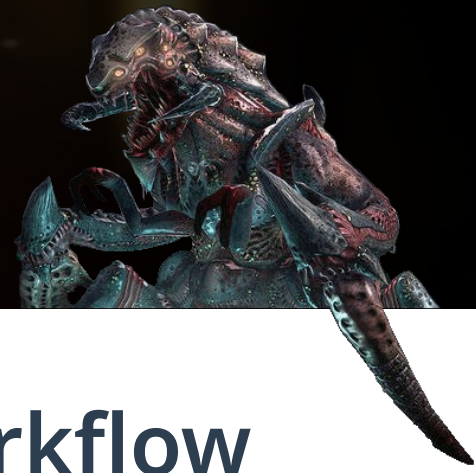
On software development and child sacrifice



Forking is like giving birth to a baby:

- **the fork cannot do anything by itself other than dying**
- **if you don't give enough care, it dies**

On software development and child sacrifice



id Software game development workflow looked like child sacrifice:

- previous one **killed to redirect resources** on the new one
- **doesn't let the child reach adolescence** and become an adult

On software development and child sacrifice



- no reusable engine
- no reusable editor
- editor **not considered as a product**

On software development and child sacrifice



Avoid situations requiring solving

- software distribution problems
- **format incompatibilities**
- **file conflicts**

On software development and child sacrifice



To solve problems

- you have to **fork your own game**

On software development and child sacrifice



Genius skills of John Carmack kept id Software alive despite

- marketing mistakes
- **strategic misses**
- opportunity losses

On software development and child sacrifice



Code was dead so

- it can be opened
- it **produces zombies**

On software development and child sacrifice



The less you get contribution back
the easier you can review

- **Opened because less open to contributions?**

On software development and child sacrifice



In 2020

- **No one wants to maintain an engine alone**
→ **Dæmon aims to be reusable**

Unvanquished: Building a community as a service



Code induced mindset

Code induced mindset



Example: PK3 VFS

- designed to receive updates coming from the seller
- **not designed for modding**

Code induced mindset



Example: DPK VFS

- a 20 years old build can load it if renamed
- loads predefined selection of assets
- turns packages into libraries
- library means suppliers means economy
→ introduces a new mindset

Code induced mindset



Example: BSP (again)

- one image format
- one image size
- **no bit to versionize**

Code induced mindset



Example: BSP with external images

(very old change)

- allows every formats including future ones
- allows every size
- **no format compatibility breakage**

Code induced mindset



id Software followed modern linear consumption pattern:

- production → consumption → trash
- **we need to recreate a cycle**
to get contributions back

Unvanquished: Building a community as a service



On open source code and sharing human communities

On open source code and sharing human communities



Dave Airlie article on building communities

tl;dr there is a big difference between open source released and open source developed projects in terms of sustainability and community.

airlied.blogspot.com

I'll start by saying it's not always a bad idea. In theory it might be possible to produce such a stack with the benefits of open source development model, however most vendors seem to fail at this. They see open source as a release model, they develop internally and shovel the results over the fence into a github repo every X weeks after a bunch of cycles. They build products containing these open source pieces, but they never expend the time building projects or communities around them.

As an example take AMDVLK vs radv. I started radv because AMD

On open source code and sharing human communities



Dave Airlie article on building communities

As an example take AMDVLK vs radv. I started radv because AMD had been promising the world an open source Vulkan driver for Linux that was shared with their Windows stack. Even when it was delivered it was open source released but internally developed. There was no avenue for community participation in the driver development. External contributors were never on the same footing as an AMD employee. Even AMD employees on different teams weren't on the same footing. Compare this to the radv project in Mesa where it allowed Valve to contribute the ACO backend compiler and provide better results than AMD vendor shared code could ever have done, with far less investment and manpower.

airlied.blogspot.com

On open source code and sharing human communities



Dave Airlie article on building communities

- **Companies have to spend** time building projects and communities around their projects

On open source code and sharing human communities



Frontier is not between humans
but **goes through each human beings**

- Same for companies

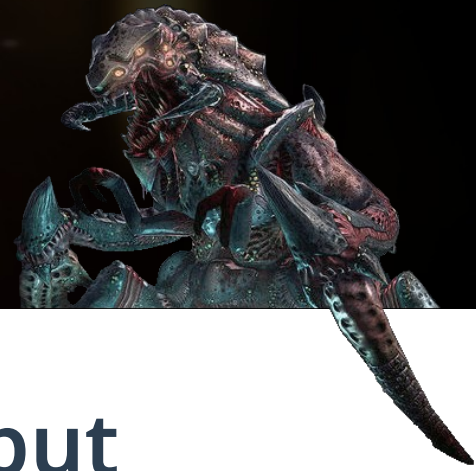
On open source code and sharing human communities



Frontier is not between humans
but goes through each human beings

- Same for companies
 - Valve or Epic engines are not free but they build communities
 - **id Software opened engines but did not build communities**

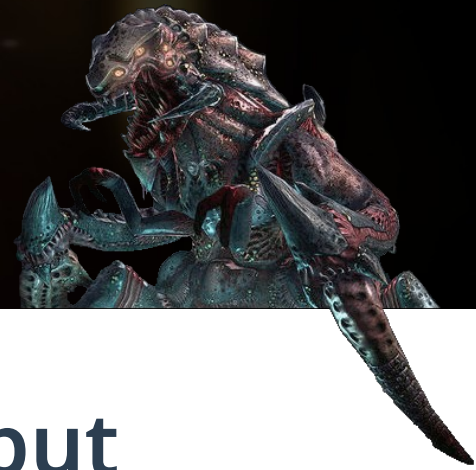
On open source code and sharing human communities



id Software technology is efficient but

- sometime the **only social relation** may have been **the sale**

On open source code and sharing human communities



id Software technology is efficient but

- sometime the only social relation may have been the sale

After Rebecca Heineman's experience,
code does not reflect methodology shift

On open source code and sharing human communities



id Software technology is efficient but

- sometime the only social relation may have been the sale

Design does not reflect problems faced by communities

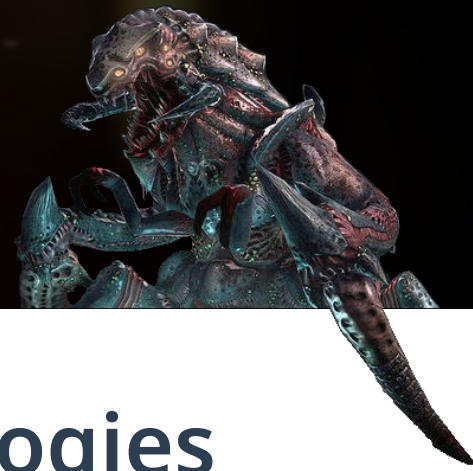
On open source code and sharing human communities



Dave Airlie article on building communities

- open source release model
 - closed source release model
- not a sharing development model

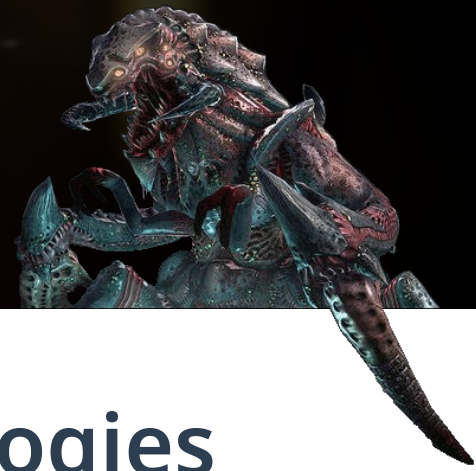
On open source code and sharing human communities



Cost of opened id Software technologies

- designed to solve technical problems
- **not designed to solve social problems**

On open source code and sharing human communities



Cost of opened id Software technologies

- designed to solve technical problems
 - not designed to solve social problems
- saves resources at the local scale
- **wastes resources at the community scale**

On open source code and sharing human communities



Cost of id Software design

- **harms growth** when software has to survive one man or a company

Unvanquished: Building a community as a service



Building a community and an ecosystem as a service

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- Human beings need to **experience existing projects as upstream**
→ no NetBlender, DarkBlender, ÜberBlender

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- Human beings need to **experience existing projects as upstream**
- People have to hear about **intention to review and merge**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- Human beings need to experience existing projects as upstream
- People have to hear about intention to review and merge
 - Android and Linux reveals **this is hard even when done right**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- **Lurk and live within communities**
- Notice aspirations, accompany births
- Like probing a market but being a friend

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- **Example:**

XreaL-based Kingpin Q3 total conversion

→ **needed a decade to reach the community**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- Talk with others even if they don't use the software
 - to open doors
 - you have to look at people's path**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- **Any patch solving a problem is for real**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- Any patch solving a problem is for real
- **There is no such thing than a finished code**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- Any patch solving a problem is for real
- There is no such thing than a finished code
- **Redesign can build up a new mindset**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- Dæmon engine is meant to be repurposable
→ help by Xonotic
to **make it a product by itself**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- Make editor more accessible to newcomers and **make workflow more natural**
→ joint effort with Xonotic

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- You can't prevent forks anyway

~_ (ツ) _/_

~_ (ツ) _/_

~_ (ツ) _/_

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- You can't prevent forks anyway
 - **Cost of forking is higher than contributing back**

Building a community and an ecosystem as a service



Making our ways through the hell of fork mindset

- We can't do magic and have to rely on good will
 - letting a child grow means everyone is free

Building a community and an ecosystem as a service



Speaker:

Thomas “illwieckz” Debesse

A service is:

- An act of being of assistance to someone
→ **first service anyone gets in life is a community**