

We are the New Guardians of the World



Tony Stanco

FreeDevelopers.Net



FreeDevelopers.Net

Hello everyone.

My name is Tony Stanco.

I'm a securities attorney who recently left my job with the Securities and Exchange Commission's Internet and Software group, to start *FreeDevelopers.Net*.

FreeDevelopers is an international, self-regulatory organization for the development of GPL software.

We already have more than 800 members from about 50 countries. It's still in the developmental stage; we are not completely up and running yet.

2

Issues and Handouts

What I am going to talk about is:

- Why is the GPL so important?
- What is Free Software really all about?
- the efficiency of Free Software development.
- the problem of how to pay Free Software developers.
- What *FreeDevelopers* is doing to solve that payment problem.

I have some handouts that cover most of what I am going to say. So, if I'm not clear on something, it may be clearer once you read those. If we run out, contact me at Tony@FreeDevelopers.net and I'll give you citations on our site for these articles. You can use that email also if you want to reach me later, too.

This is the first time that I am taking the show *on the road* (so to speak) and the presentation is still pretty rough. So it is probably going to be quite painful for both of us.

Sorry to do it to you; but *someone* has to be the first.

3

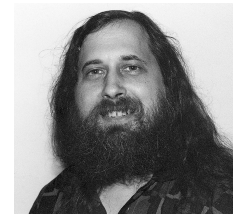
What is the GPL?

All right, let's start.

What's the GPL and why is it important?

Most of you know the story about Richard Stallman, I'm sure.

... how he created the GPL as part of the Free Software Movement and GNU Project in the early 1980s.



As most of you know, he was upset about how proprietary software companies would raid the universities, taking students and the code. They would then close the code up and thereby destroy the developer community that existed between the developers inside the company and those outside the company.

In rebellion against that all-too-common occurrence, he started the Free Software movement.

4

Free Software movement

The Free Software movement

- This is based on sharing the code and creating communities, not hiding the code and dividing them.
- He envisioned developer communities around the world where developers worked together like they did in the universities.
- He believed that the code should belong to *everyone*, so that anyone could see, study and modify it.
- He basically believed that these were *fundamental rights* that each developer had, for personal self-expression and freedom.

If you think about it, a developer can't really be a developer without access to source code. That would be like a painter without access to paint.

5

The GNU GPL

You need the access to the code to fundamentally do your job.

This access is so important that Richard wrote the General Public License (GPL), which is a key constitutional document for creating these sharing communities that he envisioned.



Most of the good things in the community (I believe) can be traced back to that single document, since *all* of the other licenses allow proprietary closure and control in one way or another.

So those other licences are not viable, in the long term, as pillars to keep the code free for the community—in my opinion.

6

Rights and Freedoms

So what?

So is that what all this is about: developer rights and freedoms?

Most people involved in this think so.

But this movement is actually *much bigger* than that.

It is fundamentally about the rights and freedoms of *everyone*.

As we enter the 21st century and technology touches the lives of more and ever more people, as interacting with others and with the world becomes more and more interacting through technology, ...

... we realise that this movement is really about basic freedoms of *everyone*, not just of software developers.

7

1st Line of Defence

Developers will just be the first line of defence to protect everyone's freedoms; but it is not just about developers' freedoms.

With the connectivity of the Internet and cyberspace, software is the functional equivalent to law in real space, because it controls people, just like law does.

But while law uses a human police force to enforce its rules, software uses a digital police force to enforce its rules.

That actually makes the digital police force *much more obedient* and therefore *more dangerous* in the wrong hands.

8

far-fetched nonsense?

Is this all far-fetched and complete nonsense?

- Look forward 10 or 20 years and imagine how much technology is going to impact the way people live.
- Think about the interconnectivity of the Internet, e-commerce, e-government, wireless connectivity, ubiquitous digital cameras, biometrics, etc.
- Technology is going to be everywhere and intermediating almost everything that people do.

Now imagine that such a world has a Hitler, a Stalin, or a Mao in power. Imagine how much more efficient they could have been if they lived in such a highly interconnected technological world?

They caused enough damage to their own citizens and others relying mostly just on paper and pencil and willing people.

9

think seriously about it

Now Free Software is not going to prevent having this technologically dependent future. That is going to happen regardless.

So this future will be bad enough even *with* Free Software.

But it will be a *total nightmare* with closed, proprietary software.

That's why Free Software is important.

You can't have a world where software is law and a digital police force is everywhere, and have that law secretly created by just a few people, and with no one else knowing what it does.

The world has fought a great many wars to make regular law open, democratic and available to those governed.

We shouldn't let technology take us backwards to a time when a few people have the arbitrary power to create whatever law they please, and have everyone else just subject to it.

I have found that technologists discount these ideas more than they should. You should *seriously think about them*.

10

Guardians of Freedom

History is full of past tyrants who would have loved to be living in the age we are now entering. Secret code and a digital police force would have made their life so much easier in tracking, discrediting and capturing dissents. With the FBI's *Carnivore* system, we are starting to see the very beginnings of this already.

In those past instances, it was some technologists who were the most vocal in cautioning against this.

That is as it must be, since it is only *you* (the technologists) who understand what the code does.

Only *you* can warn the world when that code goes too far.

So developers have now become the new "guardians of freedom" for everyone.

11

Moral Obligations

- You need to start taking your moral obligations more seriously.
- To be able to perform that function, you need to *see the code*.
- So that is why Free Software is very important stuff.
 - It is *not* just about developer rights.
 - It is *fundamentally important* to a free people and to a free world.
- You should remember that for evil to triumph, good people just need to do nothing.

12

protect everyone's rights

All right, I doubt that I have convinced anyone here about this. I seldom do.

So let's assume that all of what I've just said is complete nonsense, and that Free Software is really only about developers and developer rights and freedoms.

After all, traditionally this is what people think, when they think of free software.

Still, even then you should agree with Richard Stallman and understand the need to protect the rights of software developers.

And that will be fine; because in protecting developers' rights, you'll protect the rights of everyone else too.

13

Not just efficiency

... and that's where 'Open Source' goes wrong.

They think it is only about efficiency and better code, and that is what it's fundamentally about. Certainly our job is made a lot easier because Free Software is more efficient and produces better code. But even if that wasn't true, it would still be right to fight for it.

That's what Stallman is saying, and he's *absolutely right* about it.

Otherwise it is like saying that if tyranny is more efficient than democracy, then tyranny is better. And that's *just plain wrong*.

There are other more important issues besides efficiency: freedom, justice, equality, liberty and democracy. So even if proprietary was shown to be more efficient, you *should still fight* for Free Software.

This is mostly hypothetical because it turns out that Free Software is more efficient than proprietary; so in practical terms it doesn't really make a difference. But you should understand the difference anyway.

14

Historical analogy

Historically, this is not unlike the debate between having democracy or a nobility (oligarchy). For a very long time it was believed that a democracy was an inefficient system; that ruling élites were better than the noise from the common people.

For hundreds of years, people believed that democracy was being purchased at the cost of efficiency. Yet some people still thought it was the right thing to do.

Now it turns out, after 200 years of American history, that democracy was not only fairer, but also more efficient as well. That is why other systems collapsed, or don't produce the innovation that this country (the USA) does.

But that was only realized for sure at the end of the 20th century.

Before that it was an open question and a big experiment. So there was a moral decision made before that. It didn't matter if democracy was less efficient, it was still believed to be *the right thing to do*.

15

Freedom begets efficiency

But freedom always unleashes great energies and gives unexpected results that an unfree people don't produce, so it turns out that it is also efficient.

Still, you need to know that freedom is not about efficiency; it is about morality and fundamental justice.

We need to remember that if we are willing to sell freedom for any price, *we will never be able to buy it back later*.

It just doesn't work that way.

After all this, I'd like to now move on to talk about the efficiency of Free Software. By the way, most of this is in my article, "[Why Microsoft is a Dinosaur](#)"; so if you miss something here, just look at that. Also, a lot of what I was talking about before is in the article, "[Why *FreeDevelopers* used the Declaration of Independence as a model for the Declaration of Software Freedom: Is Software Law or Literature](#)".

16

Free Software will triumph

- Free software *is* a more efficient developmental paradigm, so it is going to ultimately win against proprietary.
- The world is constantly working towards more efficiency, getting more from less, and Free Software is part of that.
- So there is no question Free Software will eventually triumph.
- The only question is how fast we are going to get there. Will it be years or decades?

17

Free Software, more efficient

Why is Free Software more efficient?

- Free Software is more efficient than proprietary software because software development is an *intellectual* activity.
- Intellectual activities are more efficiently done using an inclusionary paradigm, rather than using the exclusionary one of the Industrial Age.
- The continual sharing of ideas and the interaction of minds refines the intellectual products and creates better ones.

The more minds, the better, because you don't know beforehand where the bright ideas will come from. And often they come from totally unexpected places; for example, Finland.

18

Free Software, more efficient

Good brains are more randomly distributed around the world than wealth and education is.

So a model that concentrates production on only the minds that are in Redmond, Washington is not as efficient as one that allows everyone from around the world to participate.

Dividing the community, and keeping things secret so only a small group gets to see it, is not the most efficient system.

This is why scientists publish and share ideas globally. This is why doctors, lawyers and journalists interact among their peers too.

This intellectual interaction makes *everyone* stronger.

If you can keep important pieces secret, you personally can do better, as Microsoft has shown.

But the *whole community suffers* because of it.

19

Lesson from History

I recently read that a few hundred years ago, surgeons would discover new techniques and not share them for marketing purposes, so that patients would go to them because they had more successes and the patients of others died.

Well, this behavior may have profited that one surgeon, but it harmed other surgeons and other patients at the same time.

And if you think about it, it actually harms that one surgeon too. He loses the feedback loop that could further improve his techniques, if others were allowed to modify and add their further innovations, as well.

20

Inclusionary not Exclusionary

We need to understand that communitarian aspects of intellectual pursuits is better; but at the same time it is completely opposite to the efficiency of private physical property, which is more efficiently produced by excluding others.

In a world of physical scarcity, it is more efficient to give people exclusionary control of physical property, so that competition enables that the limited property gets put to the highest and best use.

This exclusionary paradigm has worked well for 300 years, applied to physical property; but it is just the *wrong paradigm* for intellectual activities.

It is going from one paradigm to the other, in the case of software, that is giving most people conceptual problems.

But I think a lot of thinking people are now starting to understand that the inclusionary paradigm is *the right one for intellectual pursuits*.

21

Paying for Free Software

Even though sharing is a better, more efficient paradigm for software development, there is still one economic block to the full adoption of Free Software. While Free Software has this as a problem, it is solved quite easily with proprietary software...

The problem is:

How do you pay developers for Free Software development?

This is not really a hard issue, except that people have a hard time breaking out of their intellectual inertia.

The intellectual inertia comes from the fact that for over 300 years, commercial activity has been organized through the corporate form.

22

Intellectual Age

But the corporate form was developed for a specific purpose a few hundred years ago. That purpose was to help in financing the huge physical capital projects that were undertaken during the Industrial Age. The Industrial Age had huge projects, like railways, reservoirs, car plants, nuclear plants, etc. The investment needed for these was beyond the wealth of even the richest families. So the corporate form was invented to give limited liability to the 1000s of small investors that joined together to purchase the huge physical plants.

Now, this worked great for 300 years during the Industrial Age. But the Intellectual Age we are now entering does not have such huge capital requirements. The Intellectual Age is more about brainpower interacting over the Internet, not physical power or physical things.

This is why **GNU/Linux** could be produced without any corporate organization and with no money involved.

23

GNU/Linux

This is an amazing feat if you think about it.

It is like the flight of the bubble bee. It should be impossible. It just can't be explained under traditional business and economic theory.

Imagine, **GNU/Linux** now competes favorably with the products of Microsoft, which is considered a great corporation with huge gross margins of 80% and \$25 billion in the bank.

That was unheard of in the Industrial Age. When was the last time that Boeing had to worry that some disgruntled employees would go away and come back with a fully functional airplane that favorably competes with its own?

GNU/Linux proves that something fundamentally different is going on here. The old assumptions no longer apply and new rules need to be developed.

24

New Corporate Form

Part of those old rules and assumptions is the corporate form. The corporate organization of things has been taken for granted for 300 years. But in the new Intellectual Age it is *no longer needed*.

In fact, it is positively harmful, because the corporation tries to make everything it touches into property, even when it isn't. It also divides communities and workers into exclusive and exclusionary camps.

This is exactly what is *not needed* with Intellectual pursuits. With Intellectual pursuits, you want inclusion and sharing, so as to produce the communitarian efficiencies we talked about before.

So just as the corporate form was invented to solve the new problems that occurred at the beginning of the Industrial Age, we need a new commercial form to solve the unique problems of the Intellectual Age.

This new form needs to work with the sharing and communitarian aspects of the new age.

25

Community is the Company

At *FreeDevelopers*, we created the CommunityCompany, which I'll refer to as the "CommCo". This is an inclusive membership organization of Free Software developers from around the world.

Any developer can join. It will remain open.

The *description of the CommCo* is included in the handouts and I will not go into it here. But you can email me after you look at it, if you have any questions.

Is the CommCo the complete answer?

Probably not, but it starts us asking the right questions. The question is not "should the paradigm be proprietary or Free software?"

The answer to that question is obvious to everyone except Microsoft.

The real question is:

"How do we organize software developers, so that they can be paid to produce under a Free Software paradigm?"

26

Developers have Power

To really understand the CommCo structure, you need to understand how proprietary companies control developers, so I want to spend a little time on that.

- In software development, the developers have the natural power because they develop the code and there are relatively no physical assets that they need access to to do their work.
- This is different from Industrial companies, since the workers there need access to the physical plant to do their job; for example Boeing, again.
Boeing employees need access to the Boeing plant to do their jobs.
- So in traditional industrial companies the owners have some power by virtue of control the physical assets.

27

Power from Secret Code

But how do proprietary software companies control their workers?

- Do Microsoft employees really have to go to Microsoft to do their job?
- Proprietary doesn't have any real assets to control its workers like Industrial companies do.
- So, to get control over employees they created some, by creating intellectual property rights in secret code.

Now, you need to understand the irony here, because it is important.

- Developers are controlled by proprietary companies with chains of their own making, because they create the code that the companies then make secret to get control over them.
- And then developers can't do their job without access to the secret code, which has now magically been transformed into property that the company owns.

28

Does not break free

This reminds me of a story I once heard about a circus elephant. A visitor to the circus once notice that a huge elephant was chained by the leg to a short spike in the ground.

He couldn't understand how that little spike and weak chain could control such a huge elephant. Surely, he thought, if that elephant wanted to, it could easily break free. Quite perplexed, he asked the elephant's trainer, why the elephant didn't just break away.

And the trainer said, "the elephant doesn't know any better. It does not know that it can easily break free." That is because when the elephant was very small, they chained the elephant up the very same way. At that time, the little elephant tried and tried, but the chain and the spike were strong enough to control him. After a while he got tired of testing the chain and spike and just stopped forever. So later, when the elephant is fully grown and could easily break free, he just never tries. He has been trained to *not* break free.

And at that point, the little chain on his leg is all that's needed.

29

That little chain

To me, this is an apt metaphor as to how developers lose their natural power to proprietary corporations.

Secret code is that little chain.

- The real reason proprietary companies keep the code secret is to maintain control over the developers.
- It has nothing to do with better or more efficient code.
- Secret code is about power; maintained by dividing and subjugating the developers.

So as Richard Stallman has said from the beginning, for 17 years now:

"Free Software is fundamentally about freedom"

Please join Richard, myself and 100s of other developers in signing the "[Declaration of Software Freedom](#)" and join *FreeDevelopers*.

30