

No stakes quiz:

Identify and explain the key elements of Stallman's vision in the GNU Manifesto and the GNU General Public License.

- ▶ What **ideals** does Stallman espouse?
- ▶ What **practical measures** does he recommend/take in order to advance towards those ideals?

Make sure you identify examples/passages from the texts to support your points.

No stakes quiz:

Identify and explain the key elements of Stallman's vision in the GNU Manifesto and the GNU General Public License.

- ▶ What **ideals** does Stallman espouse?
- ▶ What **practical measures** does he recommend/take in order to advance towards those ideals?

Make sure you identify examples/passages from the texts to support your points.

After writing down responses: find someone sitting near you and explain your positions to each other. We'll discuss them in a moment.

Discuss responses for 5-10 minutes

Today's Flight Plan:

2014-10-29

└ Today's Flight Plan:

- ▶ **Foundations:** (finish) Peer production & Free software.
- ▶ **Dynamic/Challenge:** Framing a movement, establishing culture; supporting large-scale collaboration.
- ▶ **Cases:** Modularity/granularity; GNU Manifesto; GPL.

- ▶ **Foundations:** (finish) Peer production & Free software.
- ▶ **Dynamic/Challenge:** Framing a movement, establishing culture; supporting large-scale collaboration.
- ▶ **Cases:** Modularity/granularity; GNU Manifesto; GPL.

Key concepts from Benkler & Shirky readings (redux):

4 central topics

1. Industrial information economy vs. Networked information economy
2. (Commons-based) peer production
3. Failure for free
4. Granularity/modularity

└─ Key concepts from Benkler & Shirky readings (redux):

Key concepts from Benkler & Shirky readings (redux):

4 central topics

1. Industrial information economy vs. Networked information economy
2. (Commons-based) peer production
3. Failure for free
4. Granularity/modularity

Key concepts from Benkler & Shirky readings (redux):

4 central topics

1. Industrial information economy vs. Networked information economy
2. (Commons-based) peer production
3. Failure for free
4. Granularity/modularity

I asked you to choose a topic and work in small groups to:

- ▶ Describe & explain concept(s).
- ▶ Illustrate with an example
- ▶ Explain significance (in general and/or in the context of [occ])
- ▶ Suggest how to apply this concept (as a designer/advisor/builder of a community).

Key concepts from Benkler & Shirky readings (redux):

4 central topics

1. Industrial information economy vs. Networked information economy
2. (Commons-based) peer production
3. Failure for free
4. Granularity/modularity

I asked you to choose a topic and work in small groups to:

- ▶ Describe & explain concept(s).
- ▶ Illustrate with an example
- ▶ Explain significance (in general and/or in the context of [occ])
- ▶ Suggest how to apply the concept (as a designer/advisor/builder of a community).

Concluding questions for everyone:

└ Concluding questions for everyone:

► What level(s) of analysis are we at?

► What level(s) of analysis are we at?

- Both Benkler and Shirky focus on meso & macro-level analysis (groups/communities/projects and society). Both also try to show how micro-level changes/factors play out (up?) at larger levels of analysis.
- Wikipedia is a canonical example. More sociality-focused communities like the WELL seem less clear. Likewise, communities and commons occurring within the control of private organizations (e.g., Yelp). Key dimensions: open boundaries, lack of legal excludability; resource as commons (non-excludability); emphasis on non-monetary incentives; related distinctions between kinds of goods (e.g., connective vs. communal).

Concluding questions for everyone:

Concluding questions for everyone:

- ▶ What level(s) of analysis are we at?
- ▶ Which communities that we've discussed seem like examples of peer production? Which do not? How do you make these classifications?

- ▶ What level(s) of analysis are we at?
- ▶ Which communities that we've discussed seem like examples of peer production? Which do not? How do you make these classifications?

- Both Benkler and Shirky focus on meso & macro-level analysis (groups/communities/projects and society). Both also try to show how micro-level changes/factors play out (up?) at larger levels of analysis.
- Wikipedia is a canonical example. More sociality-focused communities like the WELL seem less clear. Likewise, communities and commons occurring within the control of private organizations (e.g., Yelp). Key dimensions: open boundaries, lack of legal excludability; resource as commons (non-excludability); emphasis on non-monetary incentives; related distinctions between kinds of goods (e.g., connective vs. communal).
- This is really a wide open question. Useful to analyze communities with and to try to understand the social, technological, organizational, and motivational dynamics that sustain collaboration/participation.

Concluding questions for everyone:

2014-10-29

Concluding questions for everyone:

- ▶ What level(s) of analysis are we at?
- ▶ Which communities that we've discussed seem like examples of peer production? Which do not? How do you make these classifications?
- ▶ How can you apply these principles more broadly?

- ▶ What level(s) of analysis are we at?
- ▶ Which communities that we've discussed seem like examples of peer production? Which do not? How do you make these classifications?
- ▶ How can you apply these principles more broadly?

- Both Benkler and Shirky focus on meso & macro-level analysis (groups/communities/projects and society). Both also try to show how micro-level changes/factors play out (up?) at larger levels of analysis.
- Wikipedia is a canonical example. More sociality-focused communities like the WELL seem less clear. Likewise, communities and commons occurring within the control of private organizations (e.g., Yelp). Key dimensions: open boundaries, lack of legal excludability; resource as commons (non-excludability); emphasis on non-monetary incentives; related distinctions between kinds of goods (e.g., connective vs. communal).
- This is really a wide open question. Useful to analyze communities with and to try to understand the social, technological, organizational, and motivational dynamics that sustain collaboration/participation.

For the rest of today: Free/libre & Open Source Software!

- ▶ Fogel & Stallman readings. Debian social contract.
- ▶ Focus #1 origins and history of free/libre & open source (FLOSS) software.
- ▶ Focus #2 understand the ideals and vision behind FLOSS.
- ▶ Consider: How might movement ideals/vision impact how FLOSS projects are organized?
- ▶ Contextualize FLOSS in relation to other phenomena & communities we've analyzed.

└─ For the rest of today: Free/libre & Open Source Software!

- ▶ Fogel & Stallman readings. Debian social contract.
- ▶ Focus #1 origins and history of free/libre & open source (FLOSS) software.
- ▶ Focus #2 understand the ideals and vision behind FLOSS.
- ▶ Consider: How might movement ideals/vision impact how FLOSS projects are organized?
- ▶ Contextualize FLOSS in relation to other phenomena & communities we've analyzed.



- Free software is the beginning of this story.
- Begins with the printer story. Code to run printer driver.
- In 1985, Stallman published the GNU Manifesto & GPL.
- Created a social movement.
- Created the GNU Project.
- Created the Free Software Foundation.
- Created the free software definition.

2014-10-29

Free Software Defined



Ask: So return to your no stakes quiz responses. What are the core components of Stallman's vision? How does he propose to get us there?

This is the core of Stallman's definition of Free Software: the four freedoms. Usually numbered zero-three because that's how programmer's count — they start with zero. [DISCUSS THE FOUR]. All are enacted through the GPL — **How?**

That said, this is all a bit complicated a bit specific to software.



- The freedom to **run the program, for any purpose**
- The freedom to **study how the program works, and adapt it** to your needs
- The freedom to **redistribute copies** so you can help your neighbor
- The freedom to **improve the program, and release your improvements** to the public, so that the whole community benefits



The Free Software Definition:

- ▶ The freedom to **run the program, for any purpose**
- ▶ The freedom to **study how the program works**, and **adapt it** to your needs
- ▶ The freedom to **redistribute copies** so you can help your neighbor
- ▶ The freedom to **improve the program**, and **release your improvements** to the public, so that the whole community benefits

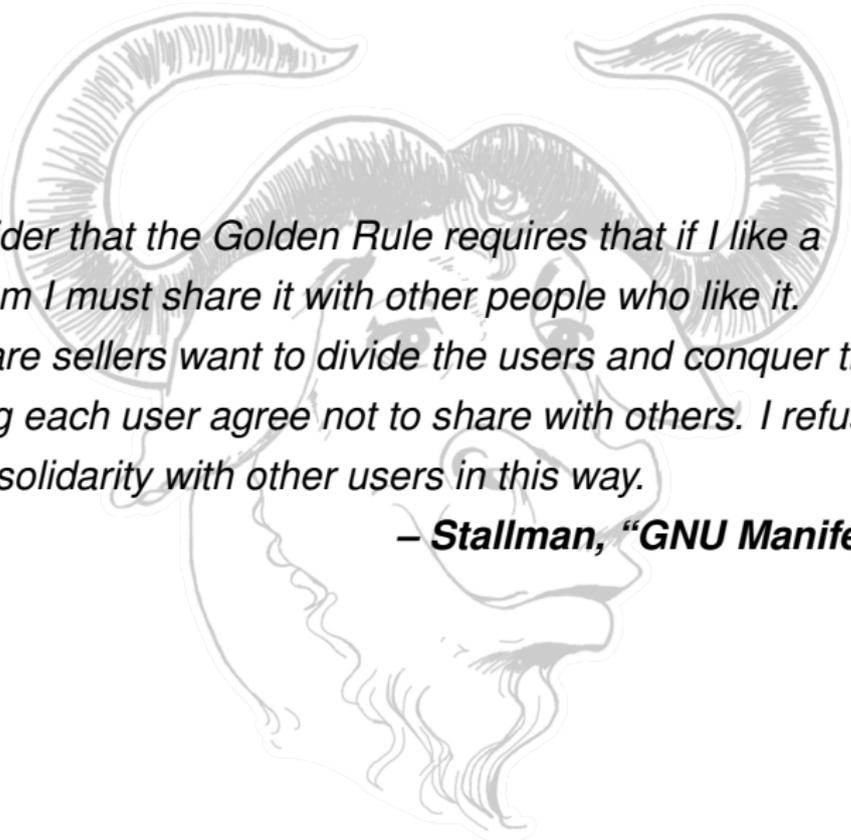


Ask: So return to your no stakes quiz responses. What are the core components of Stallman's vision? How does he propose to get us there?

This is the core of Stallman's definition of Free Software: the four freedoms. Usually numbered zero-three because that's how programmer's count — they start with zero. [DISCUSS THE FOUR]. All are enacted through the GPL — **How?**

That said, this is all a bit complicated a bit specific to software.

Stallman Key Objective #1: Access to Knowledge



I consider that the Golden Rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way.

– Stallman, “GNU Manifesto”

Stallman Key Objective #1: Access to Knowledge

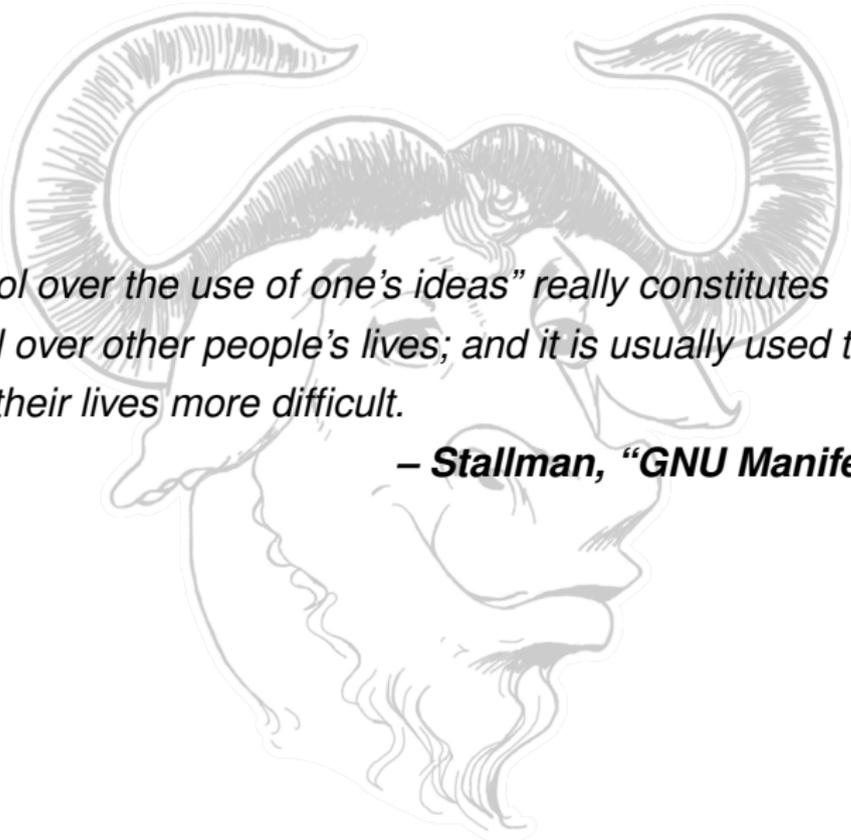
I consider that the Golden Rule requires that if I like a program I must share it with other people who like it. Software sellers want to divide the users and conquer them, making each user agree not to share with others. I refuse to break solidarity with other users in this way.
– Stallman, “GNU Manifesto”

There’s a simpler, ethical justification for access to knowledge or ideas underlying this vision and here it is.

But this is a weak form of the argument. Eben Moglen, a legal scholar and free software advocate, offers a stronger form: “In a world in which information goods have zero marginal cost, how are we supposed to justify artificial scarcity of an information good, especially one that is important?”

Again, this draws out the justification behind the **access** part of free software.

Stallman Key Objective #2: Empowerment



“Control over the use of one’s ideas” really constitutes control over other people’s lives; and it is usually used to make their lives more difficult.

– Stallman, “GNU Manifesto”

2014-10-29

└ Stallman Key Objective #2: Empowerment

“Control over the use of one’s ideas” really constitutes control over other people’s lives; and it is usually used to make their lives more difficult.
– Stallman, “GNU Manifesto”

Here’s the second theme in the four freedoms, the empowerment part: Everybody should be able to change their software to do what they want because control over software is about the capacity to exercise control over your ideas and your environment.

In this sense, the software doesn’t matter at all. It’s the people that matter. It’s about control, autonomy and power. Digital citizenship. And software is an example.

Use phone example – User Experience of the World.

The question of who controls our software is, in this sense, a deeply political question. And a deeply important question.

And Stallman’s freedoms are a an answer to the question of who should control software: users!

2014-10-29

How to get there from here? Stallman's vision of change

What practical steps does Stallman pursue/recommend in order to advance his vision of freedom?

What practical steps does Stallman pursue/recommend in order to advance his vision of freedom?



- GNU/Linux operating system
- GNU GPL – the license and the key “viral” provision (2b, 3a, 4, 5, 6).
- Free Software Foundation (supports projects).
- **Movement building around this vision of freedom**

How does a FLOSS project instantiate a Stallman-esque vision of freedoms of access and empowerment?



Describe Debian. Discuss social contract. How echoes Stallman's vision.

So what happened? How did FLOSS get big?

2014-10-29

└ So what happened? How did FLOSS get big?

In the early days, some users suffered for freedom. Things were real bad.

e.g., crashing, etc

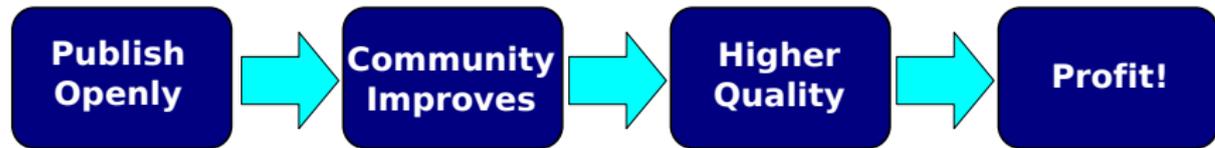
But by the late nineties, something kind of amazing started happening. The software started getting really good. A few projects, like GNU, Linux, Apache, and others, build large communities and started getting really good.

Over and over, we saw something like this.

CLICK

So what happened? How did FLOSS get big?

So what happened? How did FLOSS get big?



In the early days, some users suffered for freedom. Things were real bad.

e.g., crashing, etc

But by the late nineties, something kind of amazing started happening. The software started getting really good. A few projects, like GNU, Linux, Apache, and others, build large communities and started getting really good.

Over and over, we saw something like this.

[CLICK](#)

Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in.

– Open Source Initiative Mission Statement

Open Source is...

Open source is a development method for software that harnesses the power of distributed peer review and transparency of process. The promise of open source is better quality, higher reliability, more flexibility, lower cost, and an end to predatory vendor lock-in.

– Open Source Initiative Mission Statement

And this is basically where the word open source came in.

READ

Eric Raymond coined the term. And he did it for a few reasons:

The first reason was that it seemed to work.

The second reason was that if a person in a suit hears the word freedom, they run the other direction.

That said, the software was the same. The methodology was the same. The movements were, although with a fundamental difference in one important philosophical dimension, joined.

Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.

└─ Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.

Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.
- ▶ FLOSS projects (like Debian) and leaders instantiate freedoms through (social) contracts.

└─ Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.
- ▶ FLOSS projects (like Debian) and leaders instantiate freedoms through (social) contracts.

Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.
- ▶ FLOSS projects (like Debian) and leaders instantiate freedoms through (social) contracts.
- ▶ FLOSS also implies a *process* of software development (see Benkler's argument about modularity & granularity).

Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.
- ▶ FLOSS projects (like Debian) and leaders instantiate freedoms through (social) contracts.
- ▶ FLOSS also implies a process of software development (see Benkler's argument about modularity & granularity).

Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.
- ▶ FLOSS projects (like Debian) and leaders instantiate freedoms through (social) contracts.
- ▶ FLOSS also implies a *process* of software development (see Benkler's argument about modularity & granularity).
- ▶ Today, we live in a mixed software ecosystem. Prospects for freedoms are mixed at best (more on this later).

Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.
- ▶ FLOSS projects (like Debian) and leaders instantiate freedoms through (social) contracts.
- ▶ FLOSS also implies a process of software development (see Benkler's argument about modularity & granularity).
- ▶ Today, we live in a mixed software ecosystem. Prospects for freedoms are mixed at best (more on this later).

Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.
- ▶ FLOSS projects (like Debian) and leaders instantiate freedoms through (social) contracts.
- ▶ FLOSS also implies a *process* of software development (see Benkler's argument about modularity & granularity).
- ▶ Today, we live in a mixed software ecosystem. Prospects for freedoms are mixed at best (more on this later).
- ▶ Vision & impact of FLOSS has also spread throughout online communities (e.g., openness; freedoms; Wikipedia; free culture).

Key takeaways:

- ▶ Vision of FLOSS based on 2 ideals/promises: freedom of access; empowerment to alter code.
- ▶ FLOSS projects (like Debian) and leaders instantiate freedoms through (social) contracts.
- ▶ FLOSS also implies a process of software development (see Benkler's argument about modularity & granularity).
- ▶ Today, we live in a mixed software ecosystem. Prospects for freedoms are mixed at best (more on this later).
- ▶ Vision & impact of FLOSS has also spread throughout online communities (e.g., openness; freedoms; Wikipedia; free culture).

For Friday's lab:

2014-10-29

└─ For Friday's lab:

- Check out DQ #2 (see course schedule).
- Make sure you and your group are up to date on Wikipedia tasks.
- Come prepared to make progress on drafting your Wikipedia articles!

- ▶ Check out DQ #2 (see course schedule).
- ▶ Make sure you and your group are up to date on Wikipedia tasks.
- ▶ Come prepared to make progress on drafting your Wikipedia articles!