if moocs are the cathedral
what's the bazaar?

#lilly13 #oceb

barry peddycord iii
north carolina state university
@isharacomix

what is a mooc?

massively	what do
open	these words
online	mean to you?
course	

open mind
door
campus
access
formats
source

open source
is not about price... or licenses...

it's about a community where
everyone is invited to collaborate

basically what i'm saying is...

the role of teachers in moocs

is not only instruction & design...	but also management & maintenance

there is much that teachers can learn from open source projects

so before we get started...

what about this metaphor:
cathedral & bazaar?

what does it mean to education?
what does it mean to you?

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi) - 1991

linux was different!

source code made available
anyone could try it out
contributions were welcome
hundreds of contributors
it was really good!

how is this possible?

conventional wisdom tells us that
too many cooks spoil the soup

enter eric s raymond

the cathedral and the bazaar

foss developer since the eighties

it was a metaphor
and a case study

the creation of fetchmail
an open source e-mail service

makes nineteen claims about
open source development
- how it works
- how to make it work

don't worry - i'm not going to cover them all ;)

what can mooc teachers learn from the open source world?

starting off

[illegible]

role of students

[illegible]

collaboration

[illegible]

the endgame

[illegible]

if moocs are the cathedral

what's the bazaar?

#lilly13 #catb

if moocs are the cathedral
what's the bazaar?

#lilly13 #catb

barry peddycord iii
north carolina state university
@isharacomix

what is a mooc?

what is a mooc?

massively

open

online

course

hat is a mooc?

massively
open
online
course

what do
these words
mean to you?

open mind

door

campus

access

formats

open mind

door

campus

access

formats

source

open source

is not about price...

open source

is not about price… or licenses…

open source

is not about price… or licenses…

it's about a community where
everyone is invited to collaborate

basically what i'm saying is...

the role of teachers in moocs

is not only instruction & design...

but also management & maintenance

the role of teachers in moocs

is not only
instruction &
design...

but also
management &
maintenance

there is much that teachers can learn from open source projects

so before we get started...

what about this metaphor:
cathedral & bazaar?

what does it mean to education?
what does it mean to you?

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi) - 1991

I'm doing a (free) operating system (just a hobby, won't be big and professional like gnu) for 386(486) AT clones. This has been brewing since april, and is starting to get ready. I'd like any feedback on things people like/dislike in minix, as my OS resembles it somewhat (same physical layout of the file-system (due to practical reasons) among other things).

I've currently ported bash(1.08) and gcc(1.40), and things seem to work. This implies that I'll get something practical within a few months, and I'd like to know what features most people would want. Any suggestions are welcome, but I won't promise I'll implement them :-)

Linus (torvalds@kruuna.helsinki.fi) - 1991

linux was different!

source code made available

anyone could try it out

contributions were welcome

hundreds of contributors

linux was different!

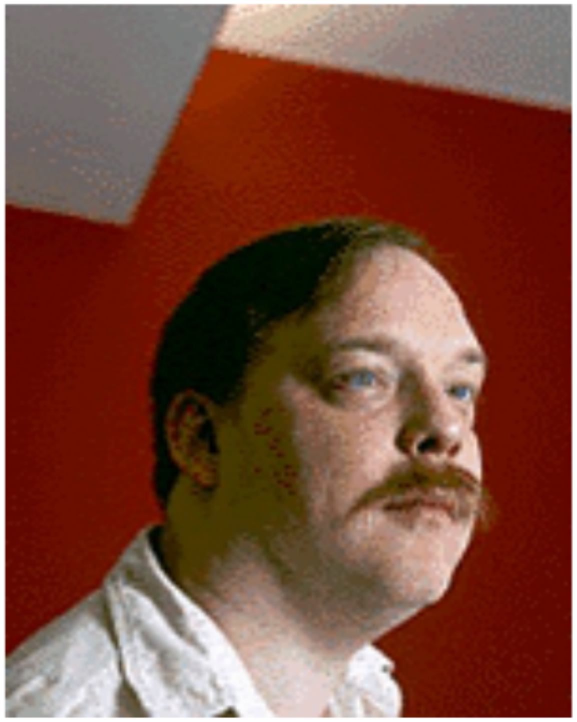
source code made available
anyone could try it out
contributions were welcome
hundreds of contributors
it was really good!

how is this possible?

conventional wisdom tells us that
too many cooks spoil the soup

enter eric s raymond

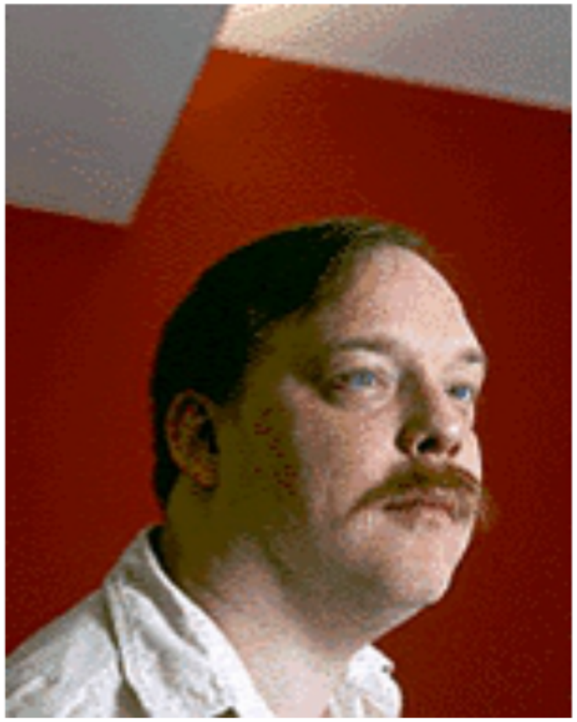
the cathedral and the bazaar



foss developer since the eighties

it was a metaphor

the cathedral and the bazaar



foss developer since the eighties

it was a metaphor
and a case study

the creation of fetchmail

an open source e-mail service

makes nineteen claims about open source development

- how it works
- how to make it work

don't worry - i'm not going to cover them all ;)

what can mooc teachers learn from the open source world?

starting off

(1) every good work of software starts by scratching a developer's particular itch
(2) good programmers know what to write. great programmers know what to rewrite (and reuse)

role of students

(6) treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging
(11) the next best thing to having good ideas is recognizing good ideas from your users. sometimes the latter is better

collaboration

(7) release early, release often. and listen to your customers
(8) given enough eyeballs, all bugs are shallow. (linus's law)

(1) every good work of software starts by scratching a developer's particular itch

(2) good programmers know what to write. great programmers know what to rewrite (and reuse)

(6) treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging
(11) the next best thing to having good ideas is recognizing good ideas from your users. sometimes the latter is better

(7) release early, release often. and listen to your customers
(8) given enough eyeballs, all bugs are shallow. (linus's law)

what can mooc teachers learn from the open source world?

starting off

(1) every good work of software starts by scratching a developer's particular itch
(2) good programmers know what to write. great programmers know what to rewrite (and reuse)

role of students

(6) treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging
(11) the next best thing to having good ideas is recognizing good ideas from your users. sometimes the latter is better

collaboration

(7) release early, release often. and listen to your customers
(8) given enough eyeballs, all bugs are shallow. (linus's law)

what can mooc teachers learn from the open source world?

starting off

(1) every good work of software starts by scratching a developer's particular itch
(2) good programmers know what to write. great programmers know what to rewrite (and reuse)

role of students

(6) treating your users as co-developers is your least-hassle route to rapid code improvement and effective debugging
(11) the next best thing to having good ideas is recognizing good ideas from your users. sometimes the latter is better

collaboration

(7) release early, release often. and listen to your customers
(8) given enough eyeballs, all bugs are shallow. (linus's law)

the endgame

(5) when you lose interest in a program, your last duty is to hand it off to a competent successor

recognize when the community doesn't need you anymore

(5) when you lose interest in a program, your last duty is to hand it off to a competent successor

recognize when the community doesn't need you anymore