

# The Cathedral and the Bazaar



**CSC 586E OSSE**

Dr. Daniel M. German

*Department of Computer Science*



**University  
of Victoria**

January 11, 2005 Version: 1.0.0

# Main observations



- Very informal paper
- But came at an important time (nobody have thought about it!)
- Sometimes it feels like preaching

# Main contributions



- First study of F/OSS
- 18 “lessons”
- Extensively cited

# Lesson No. 1



- “Every good work of software starts by scratching a developer’s personal itch”
  - Assumption, all development starts by a self need

# Lesson No. 2



- “Good programmers know what to write. Great ones know what to rewrite (and reuse)”
  - Constructive laziness
  - Reuse is certainly limited by copyright

# Lesson No. 3



- “Plan to throw one away: you will, anyhow”
  - Originally by Fred Brooks, the Mythical Man Month
  - Does it mean that F/OSS is prototyping in nature?

# Lesson No. 4



- “If you have the right attitude, interesting problems will find you”
  - Less relevant (IMHO)
  - Influenced by his FAQ on how to be a Hacker

# Lesson No. 5



- “When you lose interest in a program, your last duty to it is to hand it off to a competent successor”
  - Commercial software dies and disappears



# Lesson No. 6



- “Treating users as co-developers is your least-hassle route to rapid code improvement and effective debugging”

# Lesson No. 7



- “Release early, release often, and listen to your customers”
  - Useful in development
  - Annoying with stable versions

# Lesson No. 8



- “Given a large enough beta-tester and co-developer base, almost every problem will be characterized quickly and the fix obvious to someone.”
  - **Linus Law**: “Given enough eyeballs, all bugs are shallow.”
  - “Debugging is parallelizable”

# Lesson No. 9



- “Smart data structures and dumb code works a lot better than the other way around.”
  - Not only relevant to FOSS
  - Reminds me of “Programming Pearls”, by Jon Bentley

# Lesson No. 10, 11, 12, 13



- 10 “If you treat your beta-testers as if they are your most valuable resource, they will respond by becoming your most valuable resource.”
  - Rehashing on Lesson No 6
11. “The next best thing to having good ideas is recognizing good ideas from your users”
12. “Often, the most striking and innovative solutions come from realizing that your concept of the problem was wrong.”
13. “Perfection (in design) is achieved not when there is nothing more to add, but rather when there is nothing more to take away”

# Lesson No. 14



- Any tool should be useful in the expected way, but a truly great tool lends itself to uses you never expected.
  - The FOSS community is particularly good at this

# Lesson No. 15, 16, 17, 18



15. “When writing gateway software of any kind, take pains to disturb the data stream as little as possible - and *\*never\** throw away information unless the recipient forces you to!”
16. “When your language is nowhere near Turing-complete, syntactic sugar can be your friend.”
  - ??????
17. A security system is only as secure as its secret. Beware of pseudo-secrets.
18. To solve an interesting problem, start by finding a problem that is interesting to you.

# A response to early attacks on the paper



- “It’s fairly clear that one cannot code from the ground up in bazaar style. One can test, debug and improve in bazaar style, but it would be very hard to originate a project in bazaar mode. ... Your nascent developer community needs to have something runnable and testable to play with.”



# Lesson 0: to us



- Very subjective points of view
- Nobody has really sit down to design experiments to corroborate or disprove its points
- He is not only explaining how FOSS works, he is **preaching** how it should be done!