

The Cathedral and the Bazaar

by Eric S. Raymond

อ.ดร.ยรรยง เต็งอำนวย
Information Systems
Engineering Laboratory (ISEL)
Dept. of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Yunyong.T@Chula.ac.th
[@YunyongTeng](#)

12/13/09

ยรรยง - Cathedral & Bazaar

1

Background

- Eric Raymond, by 1993, had contributed to UNIX and open source community for 10 years.
- Big s/w should be well-planned like a Cathedral.
- Linux development model is:
 - Release early and often
 - Delegate everything you can
 - Be very open
- Linux community → a great babbling bazaar

12/13/09

ยรรยง - Cathedral & Bazaar

2

Rule # 1

- Every good work of s/w starts by scratching a developer's personal itch.
- Necessity is the mother of invention.
- Too often s/w developers spend their days grinding away for pay at programs they neither need or love.
- Linux → average quality of s/w is high.

12/13/09

ยรรยง - Cathedral & Bazaar

3

Rule # 2

- Great programmers know what to rewrite.
- Constructive laziness.
- A is for results, not efforts.
- Linus Torvalds reused MINIX codes.
- UNIX has source-sharing tradition → code reuse.
- Linux world has terabytes of open sources.

12/13/09

ยรรยง - Cathedral & Bazaar

4

Rule # 3

- Plan to throw away your s/w.
- You will not understand the problem until after the 1st time you implement a solution.
- Be ready to start over *at least once*.
- อ.ขรรชง's Law – 1-byte end-to-end

Rule 4 & 5

- If you have the right attitude, interesting problems will find you.
- When you lose interest in a program, your last duty to it is to hand it off to a competent successor.

Rule # 6

- Users as co-developers → rapid code improvement and debugging
- UNIX & Linux → users are hackers too
- Encourage users to diagnose problems & suggest fixes
- Linux Torvalds: “I’m basically a very lazy person who likes to get credit for things other people actually do.” → Lazy like a fox.

Rule # 7

- Release early & often & listen to users
- In 1991 Torvalds released new kernels more than once a day.
- Torvalds did not invent rule#7 but push this UNIX development model to the limit to match Linux complexity.

SunSITE (Sun Software, Information & Technology Exchange) is a network of Internet servers providing archives of information, software and other publicly available resources. The project, started in early 1990s, is run in by a number of universities worldwide and was initially co-sponsored by Sun Microsystems.

The more notable SunSITES include:

- SunSITE Austria [↗](#), operated by University of Vienna
- Berkeley Digital Library SunSITE [↗](#), operated by University of California, Berkeley
- SunSITE Canada [↗](#), operated by University of British Columbia
- SunSITE Chile [↗](#)
- SunSITE Czech Republic [↗](#), operated by School of Computer Science, Charles University, Prague
- SunSITE Mexico [↗](#)
- SunSITE Central Europa [↗](#), operated by RWTH Aachen, Germany
- SunSITE Poland [↗](#), operated by ICM, University of Warsaw
- SunSITE Singapore [↗](#), operated by National University of Singapore
- SunSITE Rediris (Spain) [↗](#), operated by Spanish National Research Network
- SunSITE Tennessee [↗](#) operated by University of Tennessee, Knoxville
- SunSITE Thailand [↗](#) operated by Assumption University, Bangkok

12/13/09

๒๕๕๓ - Cathedral & Bazaar

9



Rule # 8

- Given enough eyeballs, all bugs are shallow. → Linus' Law.
- Cathedral builders – few – long – arduous → disappointments
- Bazaar – many – short – bugs are shallow → more corrections ** positive attitude **

Bugs are natural – don't worry!

Debugging is parallelizable. - Jeff Dutky

12/13/09

๒๕๕๓ - Cathedral & Bazaar

11

Analogy to BT

- Rule # 8 is similar to BitTorrent phenomenon
- Popular content is quickly delivered by the aggregate bandwidth of all bitTorrent peers.
- A large content is delivered by the **Internet**
- The Linux kernel is developed by the **Internet**

12/13/09

๒๕๕๓ - Cathedral & Bazaar

12

Here, I think, is the core difference underlying the cathedral-builder and bazaar styles. In the cathedral-builder view of programming, bugs and development problems are tricky, insidious, deep phenomena. It takes months of scrutiny by a dedicated few to develop confidence that you've winkled them all out. Thus the long release intervals, and the inevitable disappointment when long-awaited releases are not perfect.

In the bazaar view, on the other hand, you assume that bugs are generally shallow phenomena - or, at least, that they turn shallow pretty quick when exposed to a thousand eager co-developers pounding on every single new release. Accordingly you release often in order to get more corrections, and as a beneficial side effect you have less to lose if an occasional botch gets out the door.

Rule # 9

- Smart data structures and dumb code work better.

Testing the Theory

- Release early & often (less than 10 days)
- Add everyone to the beta list
- Chatty announcement to beta list to encourage participation
- Ask opinion and praise beta testers

popclient

Rule # 10

- Treat beta testers as the most valuable resource
 - High quality reports
 - Bug fixes
 - Thoughtful criticism
 - Fan mail
 - Feature suggestions

Mature Bazaar-Style Project

- Beta list peaked at ~300
- People got off the list because ***the s/w is working so well.***

12/13/09

ยรรยง - Cathedral & Bazaar

17

Rule # 11

- Good ideas from others are better than your own.
- If you are honest with how you owe others, others will think you are great, e.g. Linus of Linux and Larry Wall of Perl.

12/13/09

ยรรยง - Cathedral & Bazaar

18

Rule # 12

- Innovation comes from realizing that your concept of the problem was wrong
- You are not asking the right question

12/13/09

ยรรยง - Cathedral & Bazaar

19

Rule # 13

- Perfection arises from nothing more to take away from the design
- Antoine de Saint-Exuprey (aviator & aircraft designer)
- When your code is getting both better & simpler, then you *know* it is right.

12/13/09

ยรรยง - Cathedral & Bazaar

20

Rule # 14 & 15

- Great tool has unexpected uses
- Gateway s/w must disturb data stream as little as possible.

12/13/09

มจรพจ - Cathedral & Bazaar

21

Initial Condition

- Bazaar-style project – Project coordinator
 - Plausible promise
 - Recognize good design ideas
 - Good people skills

12/13/09

มจรพจ - Cathedral & Bazaar

22

Evolution of S/W in Bazaar

- Fred Brooks's *The Mythical Man-Month*
- Gerald Weinberg's *The Psychology of Computer Programming*
 - Egoless Programming
 - Code territory
- UNIX can not do this:
 - Licenses
 - Trade secrets
 - No Internet

12/13/09

มจรพจ - Cathedral & Bazaar

23

No Internet

- Limited talent pools
 - UC Berkeley
 - AT&T's Bell Lab
 - MIT's AI Lab

12/13/09

มจรพจ - Cathedral & Bazaar

24

Linux was the first project to make a conscious and successful effort to use the entire *world* as its talent pool.

The Internet

- Anarchist's paradise
- Severe effort of many converging wills
- Principle of command vs. principle of understanding
- **Egoboo** – pleasure from public recognition of voluntary work.

Future of Open Source

Open source will win because commercial world cannot win an evolutionary arm race with open-source communities that can put orders of magnitude more skilled time into a problem.

Open source s/w is developed by the Internet.

อ.ดร.ยรรยง เต็งอำนวย