



Open Software Development
in the Real World
for Internet Developer Group 6/17/2003

W. Jolitz
President
Valux



Real World means Real Issues

- Obstacles
- Control
- Clarity
- History





Obstacles

- Mission
 - ◆ What's the point
- Politics
 - ◆ Groups & Agendas
- Liabilities
 - ◆ Patent, Copyright & Trade Secret
 - ◆ Indirect
- Financial
 - ◆ Ongoing
 - ◆ Growth

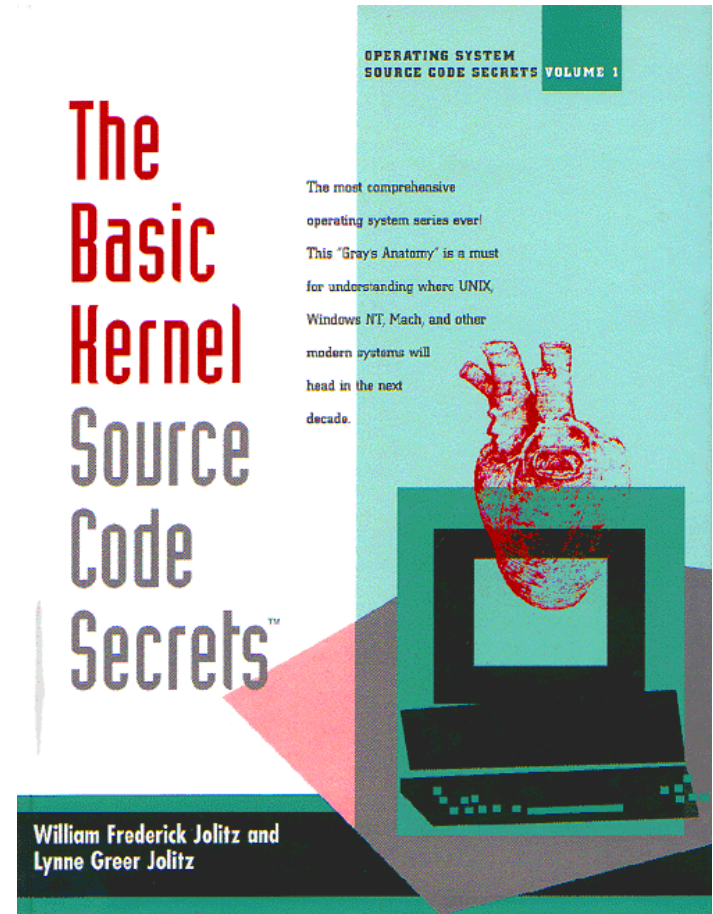


Boston Globe 6/16/2003



Control

- Loosely Coupled
 - ◆ No process
 - ◆ Frequent releases
 - ◆ Maximum speed
 - ◆ Minimum control
- Tightly Coupled
 - ◆ Grid-locked with process
 - ◆ Major schisms
 - ◆ Fewer releases
 - ◆ Maximum control
 - ◆ Minimum speed





Clarity

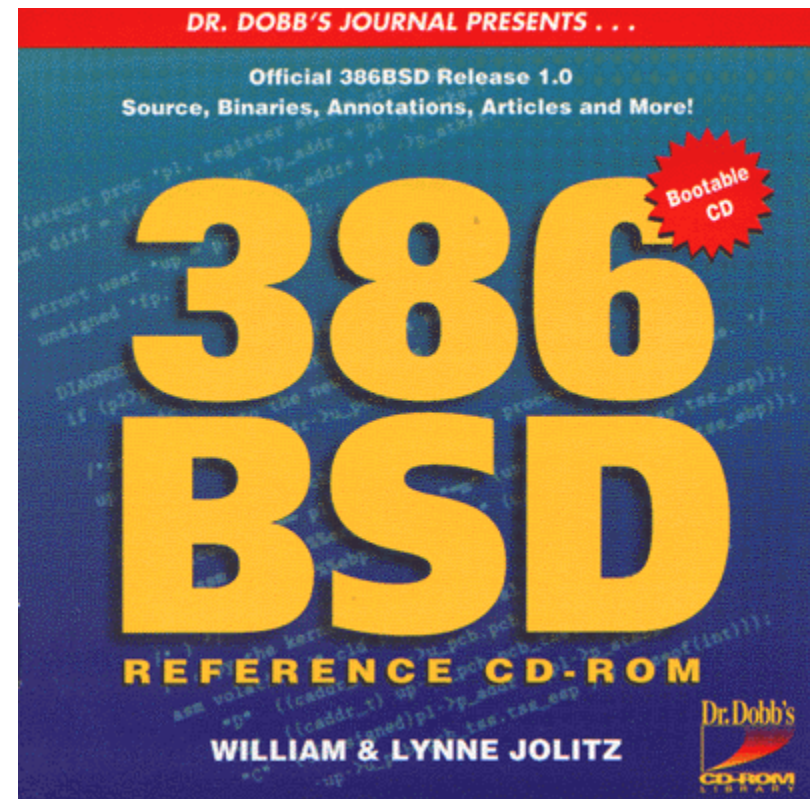
- Audience?
 - ◆ Average person
 - ◆ Enthusiast
 - ◆ Developer
 - ◆ PHD candidate
 - Goals?
 - ◆ Commercial
 - ◆ Research
 - ◆ Human factors
 - ◆ Reductionism
 - A sense of the future
-





History

- Why we were where we've been ...
 - ◆ Successes / failures
 - ◆ Prior agendas
- Why decisions were made the way they were ...
 - ◆ Expedience
 - ◆ Incremental step
- Decides new direction
- Longitudinal effects





University of California OS History

- BTSS - 60' - 70's
 - ◆ Butler Lampson et al
 - 2BSD - 70's - 80's
 - ◆ Jolitz, Kridle, Karels, Horton, ...
 - 4BSD - 80's - 90's
 - ◆ Joy, Leffler, Mcusick, Karels, ...
 - NET2/386BSD - 90's - ...
 - ◆ Mcusick, Karels, Bostic, Jolitz, ...
 - 4.4BSD/FREEBSD/NETBSD/BSDI/...
 - ◆ Mcusick, Bostic, Karels, ...
-



386BSD - Early years

- Begun in 1989
 - ◆ Original goal - move BSD to 386
 - ◆ "386BSD: A Modest Proposal"
 - ◆ Done at & in Berkeley offices
 - Used community development model
 - ◆ Driver and machine-dependant code
 - ◆ Model for following BSD open involvement
 - Documented in DDJ
 - ◆ An active account as it happened
 - ◆ Vetted code, strategy and direction
-



Early Years - Code Review Process

■ First tier - obvious

- ◆ Comments, filenames, code structure

■ Second tier - algorithms

- ◆ Classification
- ◆ Match with known targets

■ Third tier - heritage

- ◆ Developer and other works
- ◆ Progression

■ Fourth tier - hidden aspects

- ◆ Dependency on similar protected objects
-



386BSD Prior to NET/2 Release

- Match against protected code database
 - ◆ Earliest Bell Labs to latest USL
 - ◆ Other systems components besides Unix
 - ... but preserved obsolete design components
 - ◆ C-lists, B-map, cdevsw/bdevsw et al.
 - ◆ Dense memory allocation - break()
 - Later these became important
 - ◆ Where does the boundary end?
 - ◆ Are these Unix "inventions"?
-



386BSD - Post NET2

- Release 0.0 and 0.1 - Deliver on Promise
 - ◆ Basic functionality without new issues
 - ◆ Innovate stopgap mechanisms as needed
 - ◆ Dependant on Net/2 entirely for rest
 - ◆ Regulatory issues addressed by removal
 - Release 1.0 - Address Lingering Issues
 - ◆ Implement the Berkeley Architectural Review
 - ◆ Innovate active kernel object model
 - ◆ Reduce the number of redundant interfaces
 - Release 2.0 - Modular, Dynamic Kernel
-



386BSD in the Present

```
Tue Jun 10 17:07:00 GMT 2003
swapon: /dev/wd0b: device not configured
Automatic reboot in progress...
/dev/rwd0a: 5122 files, 92341 used, 97593 free (93 frags, 24375 blocks, 0.0% fra
gmentation)
starting network
add host 386bsd: gateway localhost
starting system logger.
checking for core dump...
preserving editor files
clearing /tmp
standard daemons: update cron.
starting network daemons: printer nfsiod sendmail inetd.
starting local daemons:.
Tue Jun 10 17:07:07 GMT 2003

386BSD (386bsd) (console)

login: bill
Password:
386BSD Release 1.0 by William and Lynne Jolitz.
Copyright (c) 1989-1994 William F. Jolitz. All Rights Reserved.
386BSD 1.0.220 10/03/96 20:39

386bsd 1 %
```



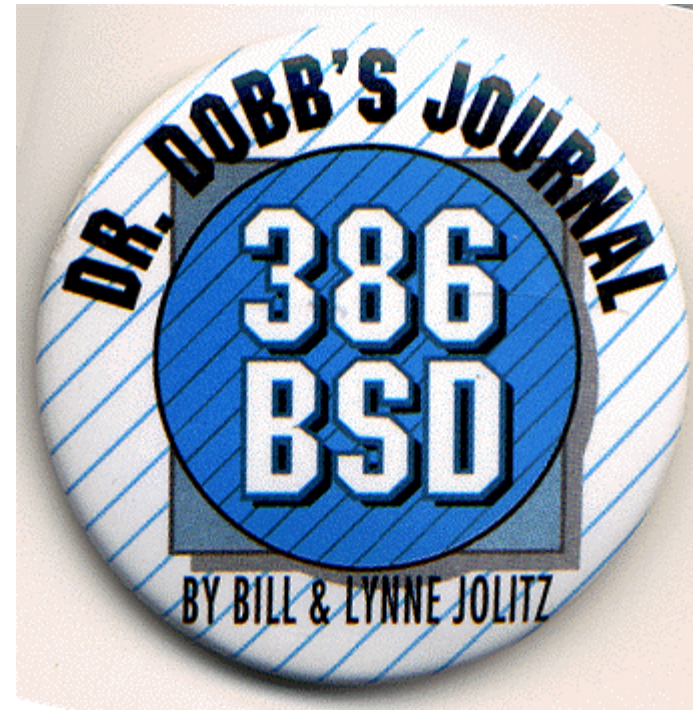
386BSD - Beyond State of the Art

- Modular - Compartmentalized
 - ◆ Components live, changed in real-time
 - ◆ Interfaces independently managed
 - Dynamic - Always current
 - ◆ Web based mechanism
 - ◆ Roll forward/back versions on fault
 - ◆ Fault injection, isolation and verification
 - One version yet many "flavors"
 - ◆ Off multi-homed web site(s)
 - ◆ Choice of many different, interoperable items
-



What's NOT important

- Licensing - GPL vs Berkeley
 - ◆ Non-issue
- Rights enforcement
 - ◆ Up to rights holder
- Unfounded litigation
 - ◆ Cost of doing anything
- Open Source Business Model
 - ◆ Necessary but unrelated





What is important

Leadership

- ◆ Must set a direction
- ◆ “Not a punching bag for ideas”

Accountability

- ◆ Admitting mistakes
- ◆ Recovery

Clear and obvious intent

- ◆ Unclear? Pay attorneys
- ◆ Clear? more developers
- ◆ Chosen by your actions





Necessary Components to Present Open Source Issues

- No serious IPR process
 - ◆ Qualified review
 - ◆ Issue list and mitigation
 - No independent authority insuring process
 - ◆ BSD had Berkeley
 - ◆ 386BSD had DDJ and publishers
 - Napster - no trust anymore
 - ◆ Rebel status now just "Anti-professionalism"
 - ◆ Tech lost its judicial advocates
 - ◆ "Small time hood" attitude unattractive
-



So when's it all going to end?



- Its another “bubble burst” ... need:
 - Dedicated, Disciplined Professionalism
 - ◆ No “over the fence” code - a joke!
 - ◆ Clearly documented work from start
 - ◆ Rapid, responsive issue mitigation
 - History of Trust and Reputation
 - ◆ Founded on institutional roots
 - ◆ Justification - papers, articles and books
 - ◆ Recognize “open-ness” requires defense
 - ◆ Proactive - not just playing pathetic
-



What are we in store for?

■ More Litigation

- ◆ Have not seen the “big guns” yet
- ◆ Expect over-reaching claims

■ Indeterminacy

- ◆ No clear time of resolution

■ Uneven treatment

- ◆ Selective targeting

■ New Development Base

- ◆ “Cheap fix” - do it from scratch with “clean team”
-



Thanks for being a great audience.



william@valux.com
<http://valux.com/wj>
