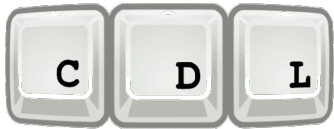


## Modelul de dezvoltare al nucleului Linux

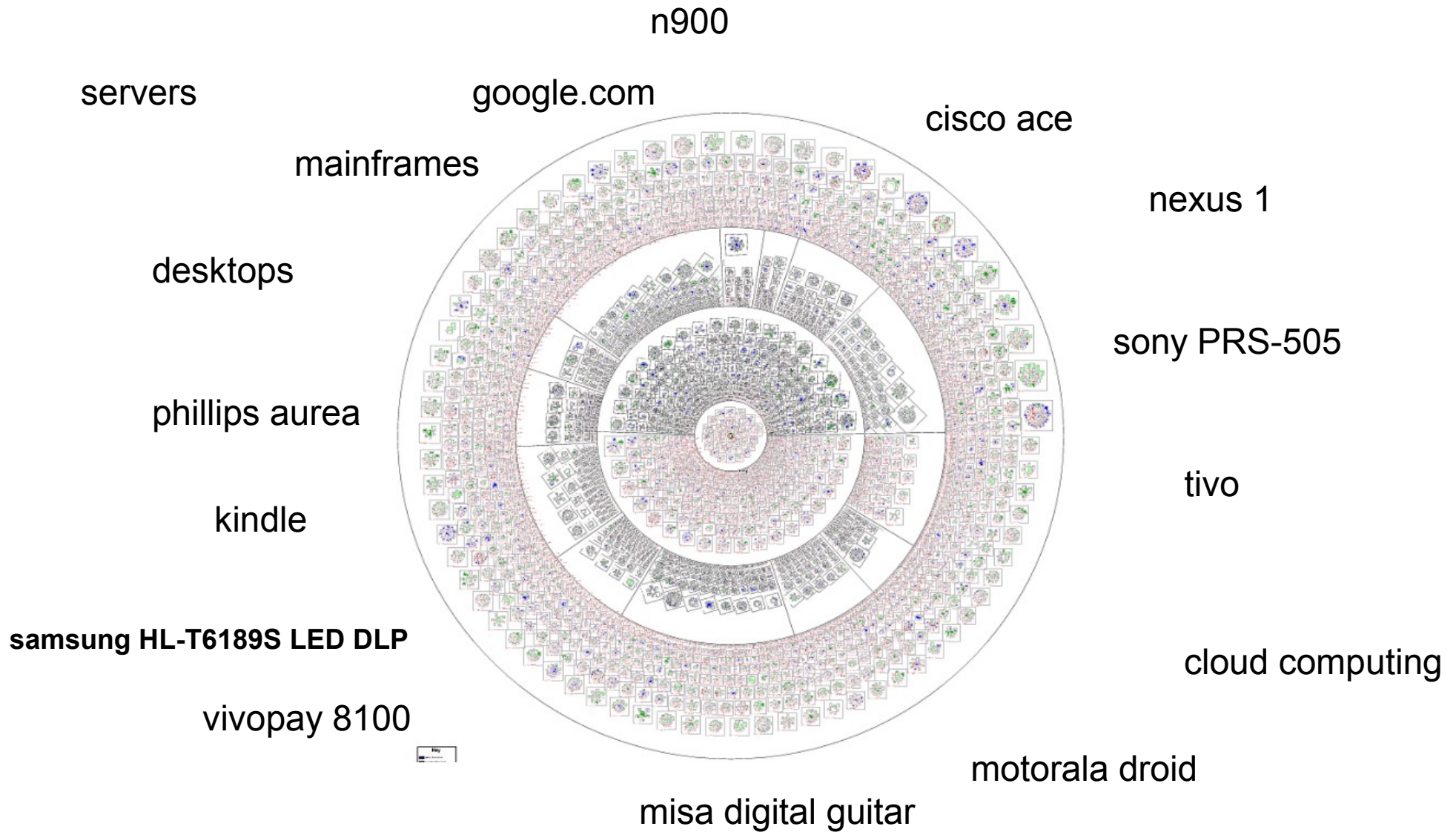
Octavian Purdilă  
[tavi@cs.pub.ro](mailto:tavi@cs.pub.ro)

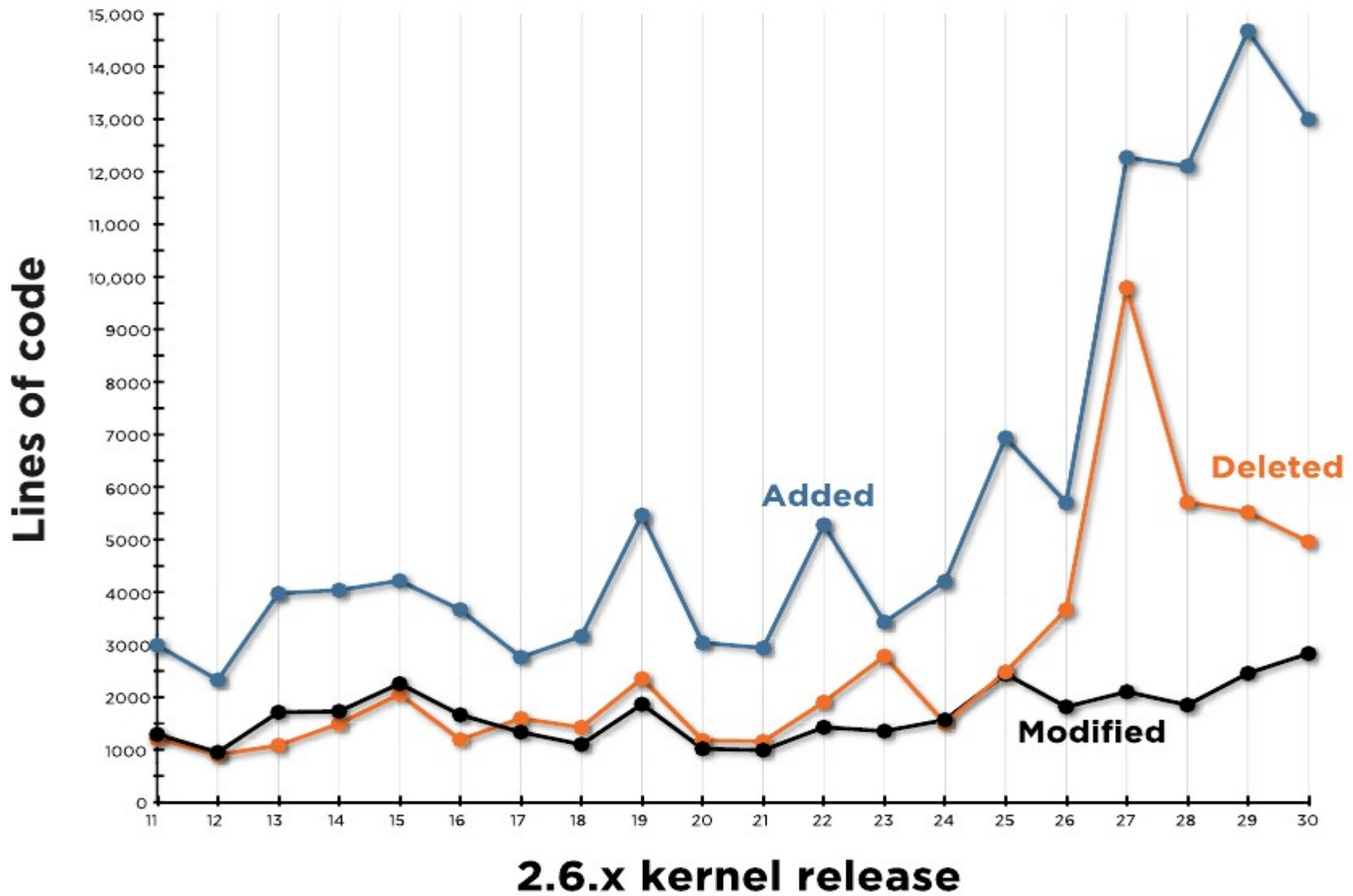
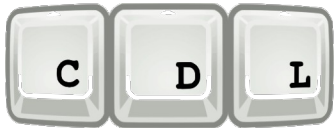
18 aprilie 2010

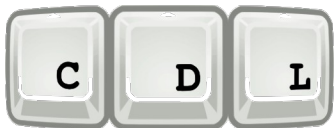




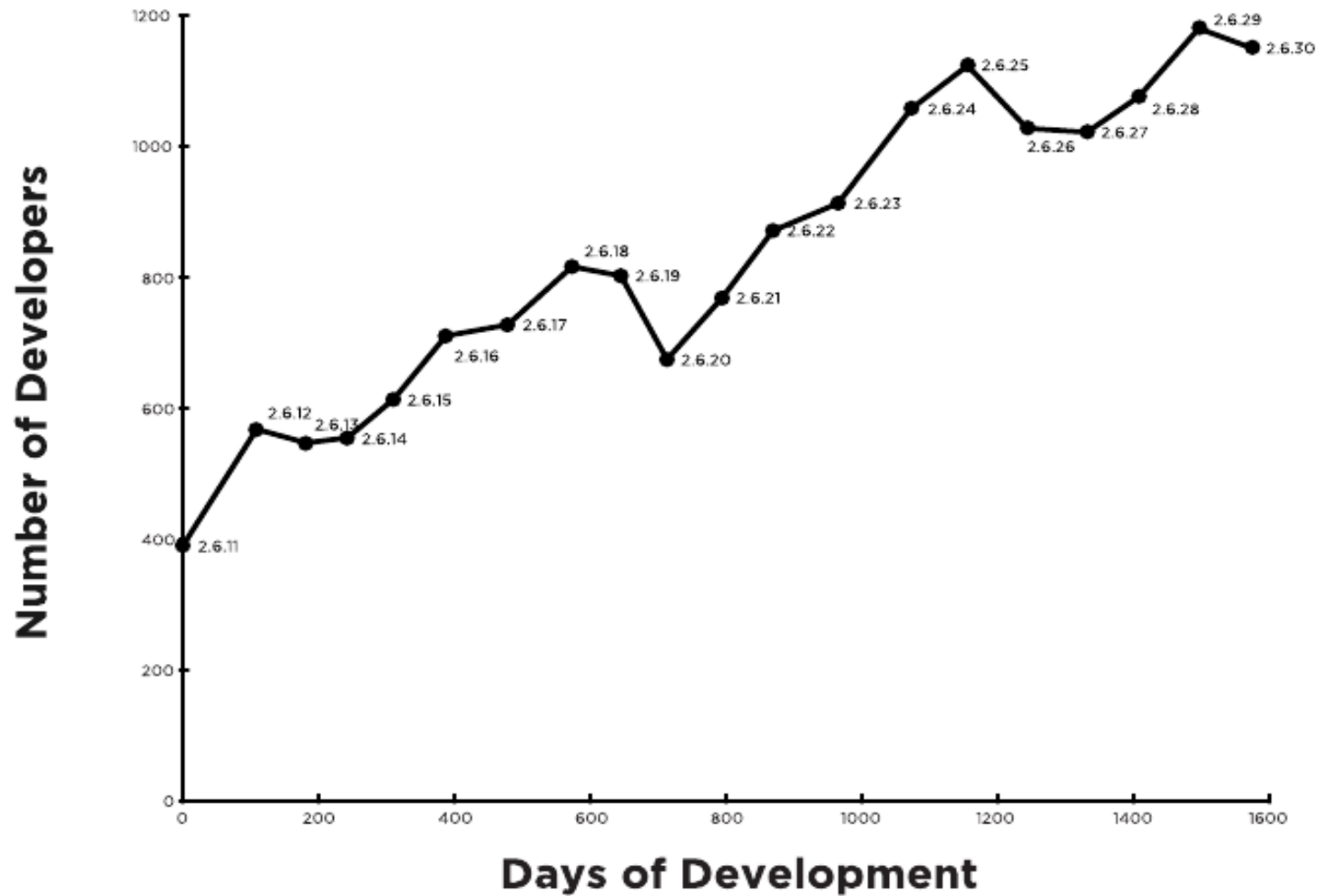
# Powered by Linux (Kernel)

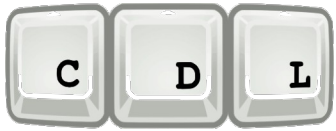






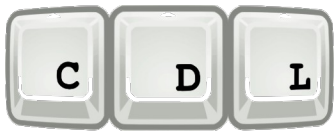
## Number of developers per release





# Linux 2.6.33

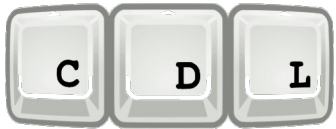
- 83 de zile
- 10,500 commit-uri
- +900,000, -520,000 linii de cod
- 1152 contribuitori
- 182 de companii
- RedHat, Intel, IBM, Novell, Nokia, Fujitsu, Texas Instruments, Atheros Communications, (Academia), (Consultants), AMD, Oracle, Analog Devices, Renesas Technology, Pengutronix, HP, Samsung, VMWare, Realtek, Broadcom, Solarflare Communications



# De ce contribuie firmele upstream ?

- Maintenance cost
- Review
- Testare
- Training
- Freebies
- Influence
- Employee recognition

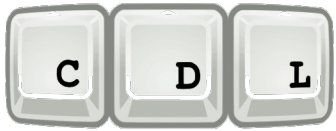




# 1810 - 1835, Cornwall, UK



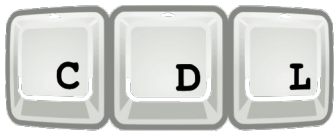
- Trevnick inventează motorul cu aburi de tip “Cornish” pe care nu îl patentează
- Un grup de manageri de mină decid să pună bazele unui jurnal tehnic în care să se descrie îmbunătățirile aduse motorului
- Diverse mine angajează ingineri pentru a îmbunătăți motorul; rezultatele sunt publicate în jurnal
- Eficiența motoarelor cu aburi crește de 5 ori față de perioada anterioară



# Modelul de dezvoltare inițial

- Două versiuni
  - Stabilă (1.0, 2.0, 2.2, 2.4)
  - Dezvoltare (0.x, 1.1, 2.3, 2.5)
- Linus este maintainer-ul pentru versiunile de dezvoltare
- Versiunile stabile au alți maintaineri (Alan Cox, Marcelo Tosati)
- Timpul dintre două versiuni stabile majore: 1-2 ani

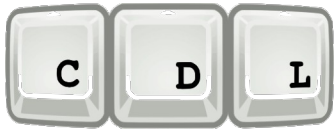




# Patch based model

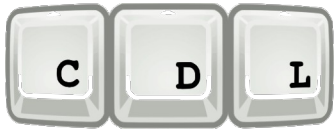
- Modificările sunt trimise sub formă de patch-uri
- Patch-urile sunt revizuite și discutate pe liste de discuții
- Maintainerii primesc patch-urile de la developeri, le testează și organizează în forma unor seturi de patch-uri pe care le trimit apoi mai departe la Linus
- Linus acceptă patch-uri direct de la developeri sau de la maintaineri
- “Linus burnouts”





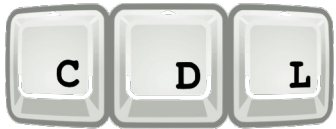
# BitKeeper

- SCM distribuit, closed source
- Folosit pentru dezvoltarea Linux începând cu 2 Feb 2002 (2.5.3)
- Fiecare maintainer menține un “bk tree”
- (maintainerii ce nu folosesc BitKeeper trimit în continuare seturi de patch-uri)
- Linus aduce în tree-ul propriu modificările direct din tree-urile maintainerilor



# Linux 2.6.0

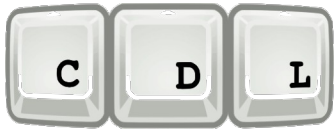
- 680 zile (17 Decembrie 2003)
- 27,149 patch-uri (1.66 patches/hr)
- 916 contribuitori
- Perioada de stabilizare
  - 2.6.0 - 2.6.8 (August 2004, 2.2 patches/hr)



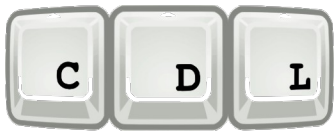
# -mm tree



- Menținut de Andrew Morton
- Folosit pentru a detecta și rezolva conflictele din timp
- Conține toate patch-urile și tree-urile bk ale maintainerilor
- Linus trage modificările direct din tree-urile bk ale maintainerilor și integrează patch-urile din -mm

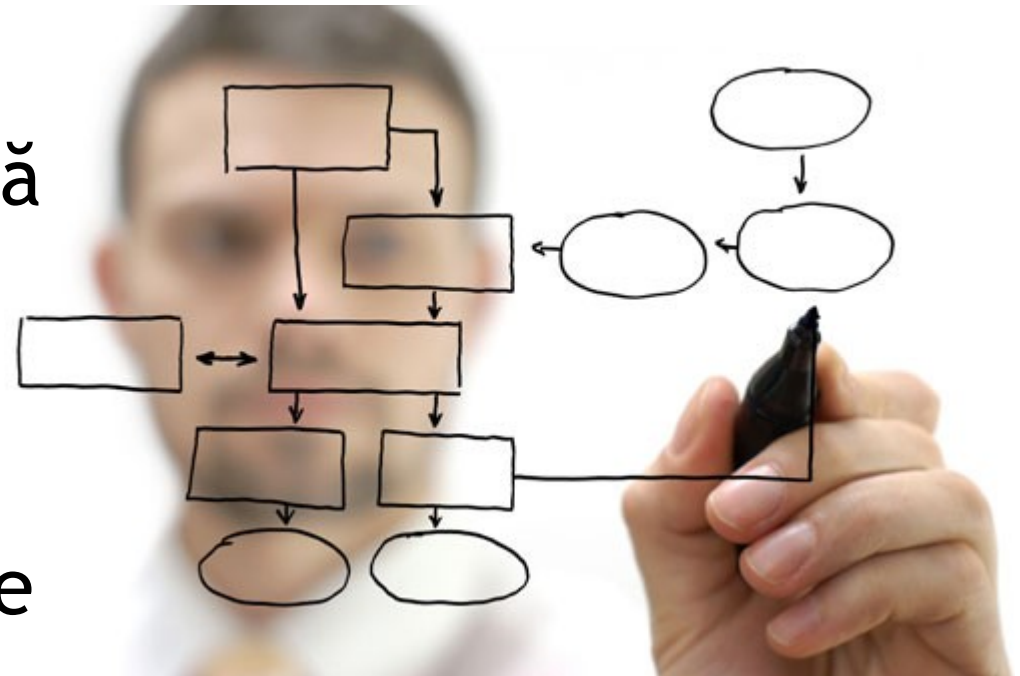


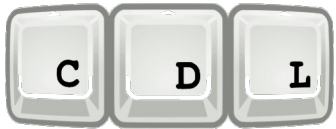
- 8 luni de “stabilizare”
- +1,200,000, -850,000 linii de cod
- ~1/3 din kernel a fost modificat
- KS2008: Se renunță la modelul în care există un branch de dezvoltare separat, toate versiunile 2.6 vor fi versiuni stabile
- Linux 2.6.x.y: doar security & bug fixes
- (stable team: Greg KH & Chris Wright)



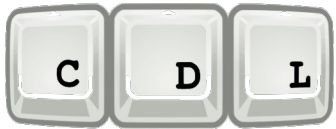
# Linux 2.6.x (x > 8)

- Merge window (7 - 14 zile)
- Se releasează -rc1, -rc2
- Urmează apoi o perioadă de ~ 8 săptămâni în de stabilizarea (bug fixing only)
- Se scoate un nou release candidate în fiecare săptămână (rc3, ..., rc8)

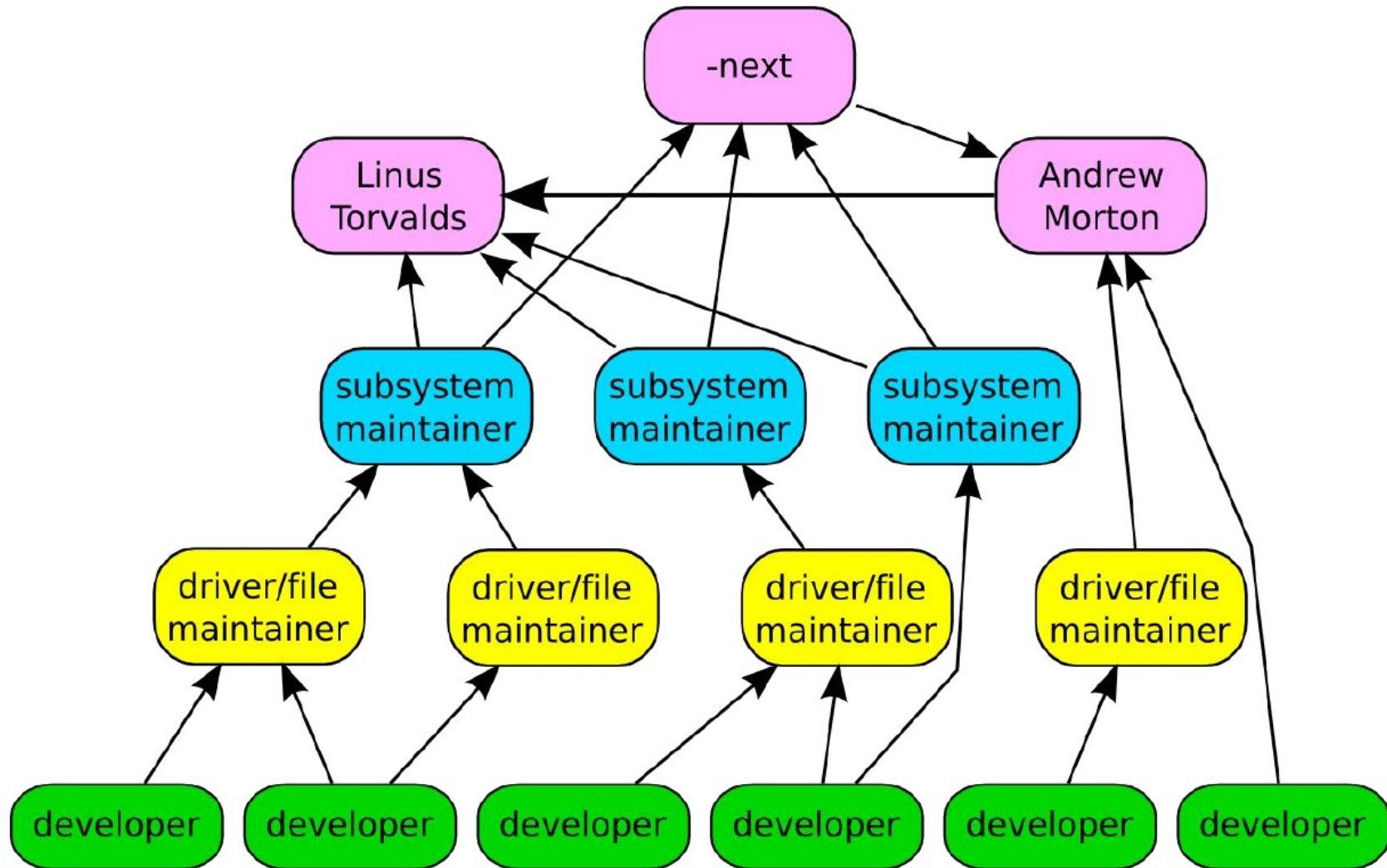




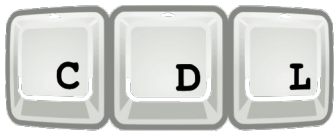
- Andrew Tridgell “dezasamblează” protocolul BitKeeper
- Larry McVoy oprește dezvoltarea versiunii free (folosită de dezvoltatorii Linux)...
- Și refuză să vândă licențe comerciale către oricine este angajat OSDL
- Linus face o “pauză” și scrie git



# Modelul curent de dezvoltare



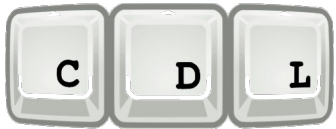




# Git trees

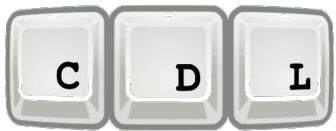
- Linus
- mm (Andrew Morton)
- next (Steven Rothwell)
- net-2.6, net-next-2.6;  
sparc-2.6, sparc-next-2.6  
(David Miller)
- ...
- 2.6.32.y (stable team)





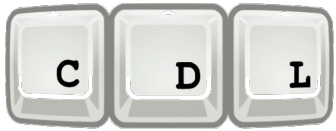
# Quality Assurance

- Bugzilla.kernel.org
- <http://kerneloops.org/> (Arjan van de Ven)
- Regression tracking (Rafael J. Wysocki)
- Tip testing (Redhat, Ingo Molnar)
- Sparse
- Coccinelle (Julia Lawall)
- Patchwork
- [scripts/checkpatch.pl](http://scripts/checkpatch.pl)



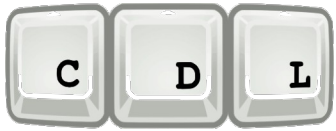
# Contributing tips

- Lurking pe listele de discuții pentru a învăța procesul (diferă minor în fiecare subsistem)
- Lasă timp pentru review (cel puțin 3 zile) înainte de a da “bump”
- Dacă lista are un site patchwork asociat, verifică starea patch-ului înainte de a face bump
- Trimite modificările într-un set de patch-uri; fiecare patch trebuie să facă un singur lucru
- Verifică coding style-ul înainte de trimitere (scripts/checkpath.pl)



## Contributing tips (2)

- When in doubt, trimite patch-uri pentru -next
- Fii responsive, mai ales cu bug-urile care îți sunt raportate
- Pregătește-te să treci prin mai multe iterații de submit, review, fix, resend
- Folosește git și branch-uri (câte unul pentru fiecare patch set pe care îl trimiți)
- Caută “aliați” pe listă și lucrați împreună



# Contributing tips (3) - Subject email

- [net-next PATCH]
- [PATCH 0/2], [PATCH 1/2], [PATCH 2/2]
- [RFC]
- [RFC PATCH]
- [RESEND] [PATCH]
- [net-next PATCH v3 0/1]

