




Linux Governance Case Study

Dr. Luis Ibanez, Kitware

© 2008 Luis Ibanez

- This presentation is Copyrighted by Luis Ibanez
- This presentation is distributed under the Creative Commons Attribution License 3.0:

<http://creativecommons.org/licenses/by/3.0/>

- You are free to Reuse 
- You are free to Remix 
- Provided that you give credit to the author 

This presentation was created using

Open Source Software



Open Office copyright is jointly held by Sun Microsystems and Contributors.

The software is distributed under the GNU Lesser General Public License Version 2.1.

Governance Models

- Roles of project participants
- Process of
 - Decision making
 - Communicating
 - Sharing

Need for a Governance Model

- Clear path for contributors to engage in the project.
- Important for sustainability.
- Sets out expectations of community members.
- States the rules for decision making
 - Preventing
 - Ambiguous situations, and
 - Subsequent chaos and cosmic destruction

Governance Models

Type	Objective	Control Style	Community Structure	Examples
Exploration Oriented	Sharing innovation and knowledge	Cathedral-like central control	Project leader, many readers	GNU, Perl, Linux Kernel
Utility-oriented	Satisfying an individual need	Bazaar-like decentralised control	Many peripheral developers, peer support to passive users	GNU Linux (excluding Kernel)
Service-oriented	Providing stable service	Council-like central control	Core members instead of project leader, many users that develop systems for end users	Apache, PostgreSQL

Governance Models

- Meritocracies
- Benevolent Dictatorships

Initial Barriers

- Perception of “Red Tape”
- Worry of losing project's flexibility and agility
- Fear of losing project's strategic control
- Too early for attracting new contributors,...
therefore, no need for rules yet.

Red Tape

- But... the purpose is not to restrict contributions
- Model must be simple but effective
- No need of having rules for **every** situation
- Lightweight way of guiding the project
 - Clear and transparent.

Loss of Flexibility

- Is actually an evidence of a poor / bad model
- A good model should
 - Increase agility
 - Guiding contributors to take new directions
 - But still aligned with the project goals
 - Provide Alignment: clear goals
 - Prevent random wandering
 - Allow complementary work

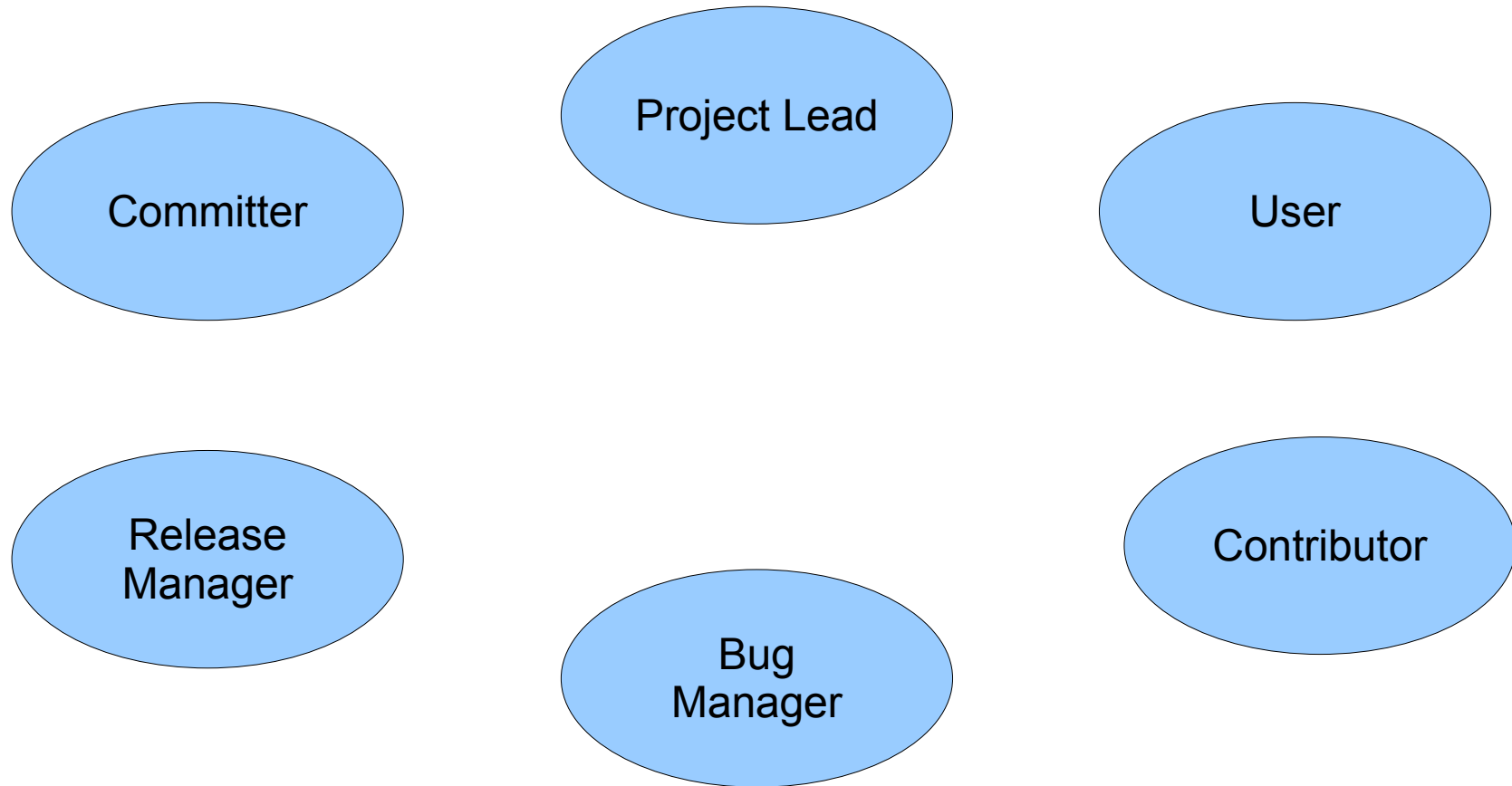
Loss of Control

- Fear of losing control to third parties
 - Due to newcomers empowering
- A good model should
 - Provide power and restrict power
- A level of leadership should always be maintained

The Project is too young...

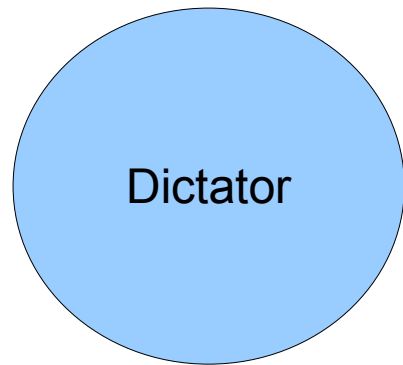
- Too young for attracting contributors
- But not too young for lacking structure
- Lack of governance model makes the project look (and be) disorganized and not very serious
- Contributors don't knock twice.
 - When they look at the project, the first impression may be the last one too.
- Never too young

Project Roles



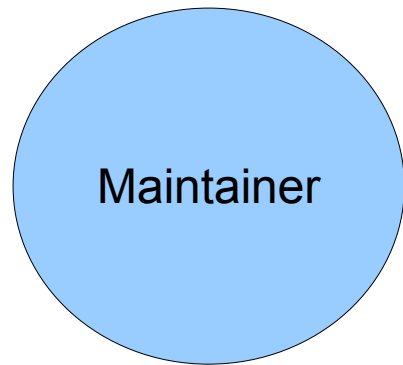
Linux Kernel

Benevolent Dictatorship Model



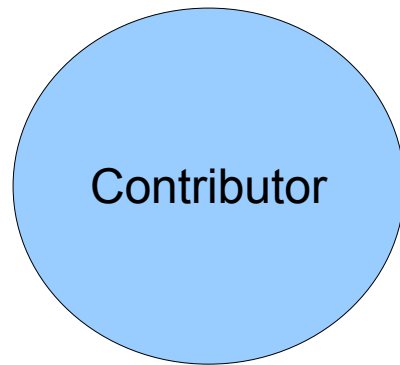
- Community building skills
- Deep technical knowledge
- Commitment and dedication
- Last word in case of disagreements
- Forking is always possible, so... the dictator must behave

Benevolent Dictatorship Model



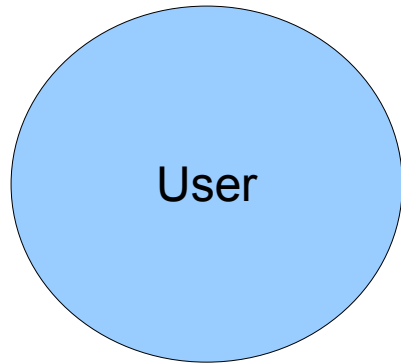
- High level of expertise
- Dedication to quality
- ...that earned them respect
- Bridge between the lead and the users
- Executing decision of the project lead
- Quality Assurance

Benevolent Dictatorship Model



- Open to anybody
- No expectation of commitment
- No specific skills
- No selection process (if maintainers have a process of code quality control)

Benevolent Dictatorship Model



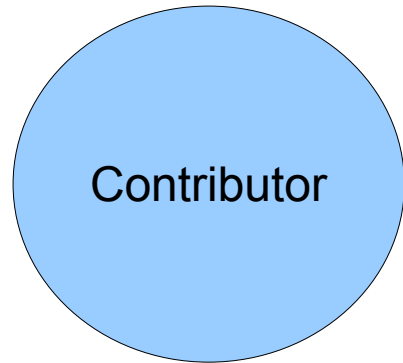
- The MOST important members of the community
- Software with no users... is pointless
- Anyone who have a need for project X.
- Moral support (developers appreciate a “thank you”)

Debian Governance

Meritocracy Model

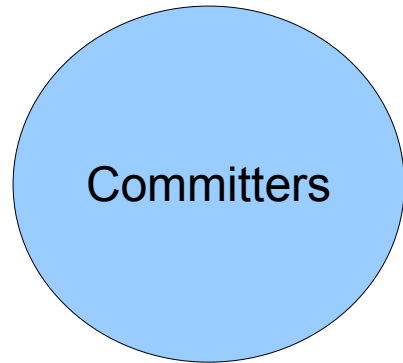
- Gives control away to community members in response to their contributions to the project
- Flat structure: all members with decision making authority have the same authority

Meritocracy Model



- Deep technical knowledge
- Commitment and dedication
- Helping others

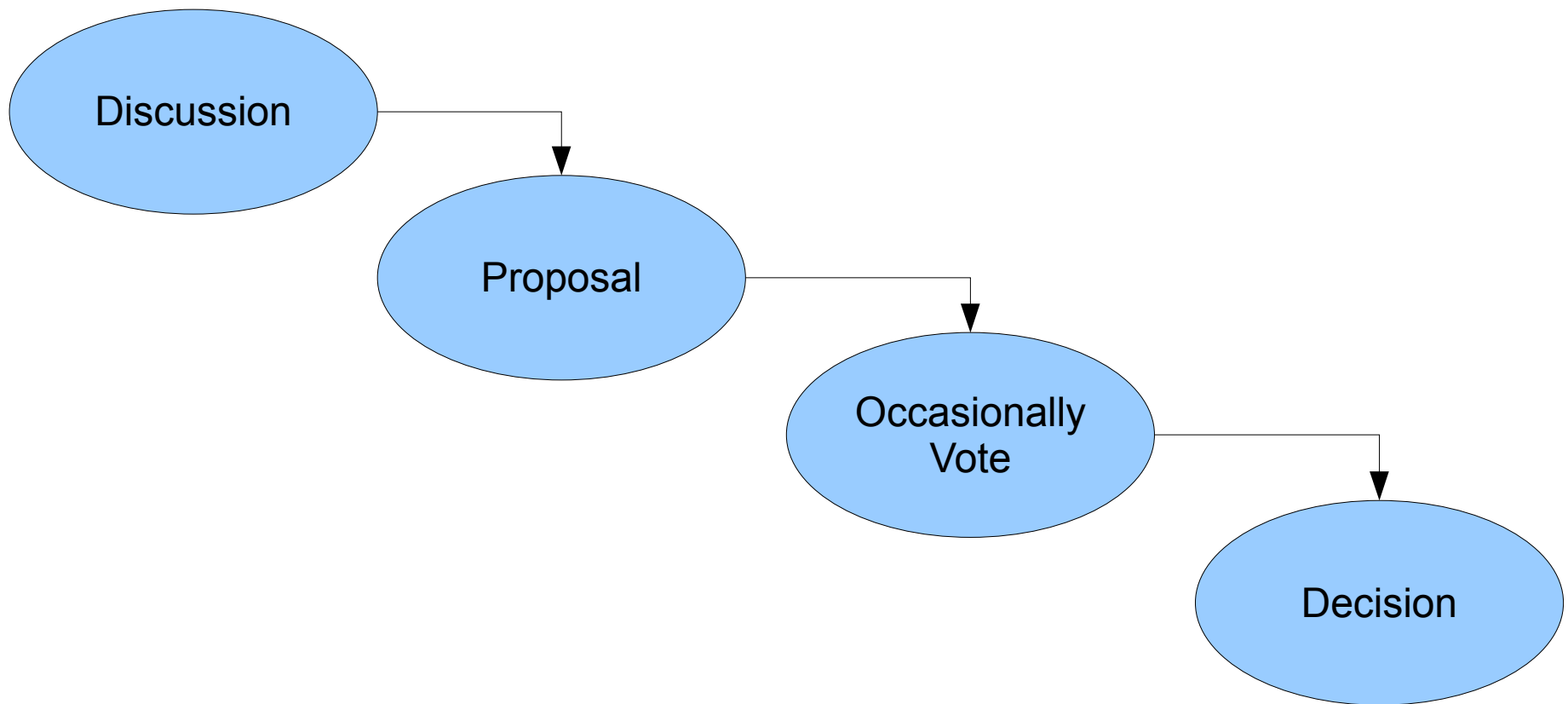
Meritocracy Model



- Receive patches from contributors
- Code review
- Commit to project
- Quality assurance

Decision Process

Lazy Consensus



Linux Kernel Summit

<http://www.deviceguru.com/16-interviews-with-linux-kernel-hackers/>

Linus Torvals

<http://www.linuxfoundation.org/events/node/154>

Ted Ts'o of (Linux Foundation/IBM)

<http://www.linuxfoundation.org/events/node/155>

Cris Mason (Oracle)

<http://www.linuxfoundation.org/events/node/149>

Greg Kroah-Hartman (Novell)

<http://www.linuxfoundation.org/events/node/152>

Paul Mackerras (IBM)

<http://www.linuxfoundation.org/events/node/155>

Dirk Hohndel (Intel)

<http://www.linuxfoundation.org/events/node/159>

Dave Jones (Red Hat)

<http://www.linuxfoundation.org/events/node/160>

End