An Empirical Study on the Structural Complexity Introduced by Core and Peripheral Developers in Free Software Projects

Antonio Terceiro
Luiz Romário Rios
Christina Chavez

LES – DCC – UFBA

LABORATÓRIO DE ENGENHARIA DE SOFTWARE

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

fapesb
Fundação de Amparo à Pesquisa do Estado da Bahia

Motivation

Developers rewriting entire systems

- EOG (GNOME's image viewer)
- GNOME-session

Why rewriting?

The code became so **complex** that rewriting pays off.

Where does all that complexity come from?

- Conventional setting: appointed designers
- Free software projects: evolutionary designs

Software Aging – Parnas (1994)

- Lack of movement
- <u>Ignorant surgery</u>

In this paper we investigate

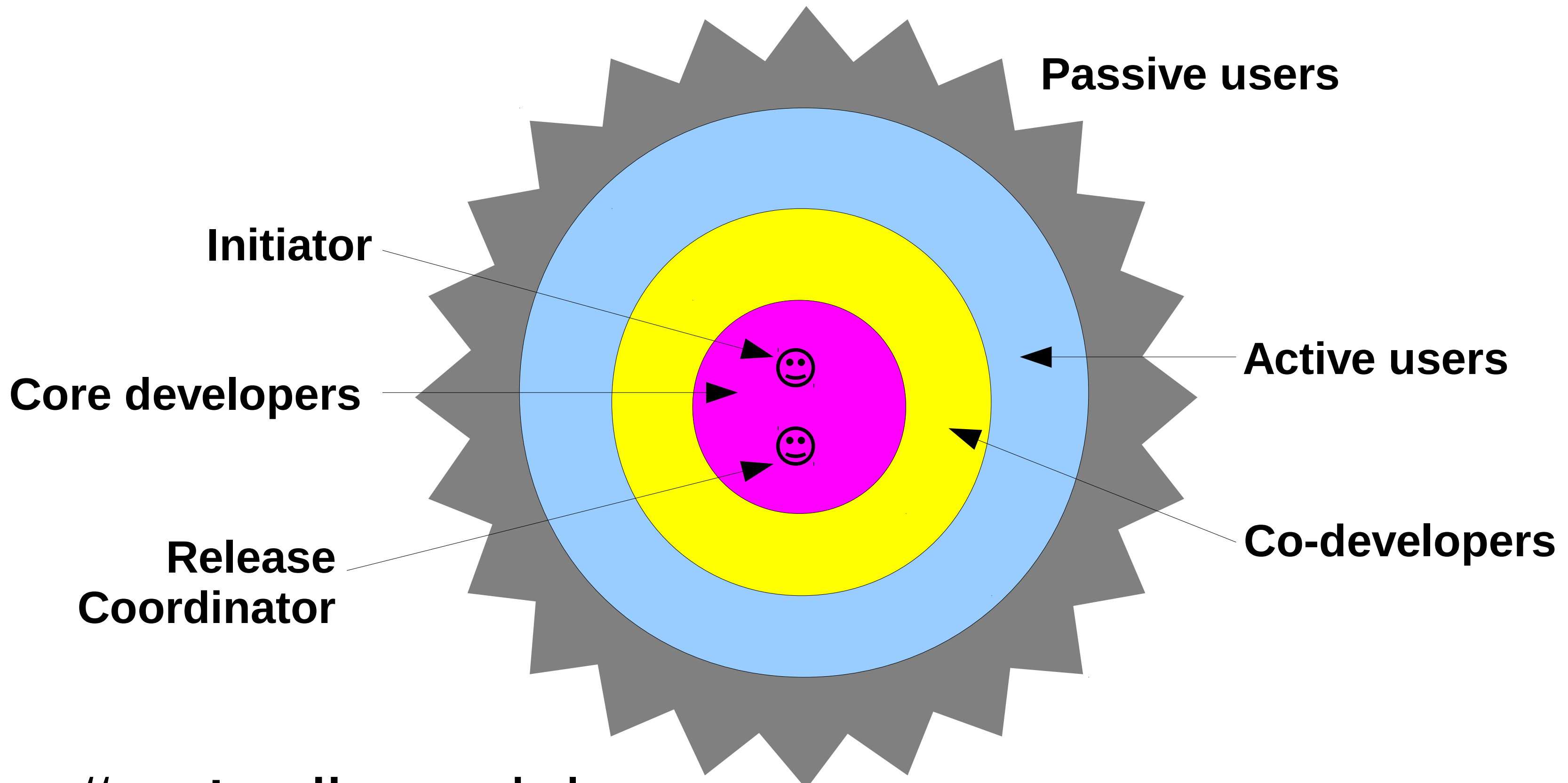
- Developers' level of participation
- Structural complexity

Background

Free software projects

- Source code availability
- User/developer symbiosis
- Non-contractual work
- Work is self-assigned
- Geographical distribution

Core and periphery in free software projects



The “onion” model.

Adapted from [Crowston and Howison, 2005]

Structural complexity

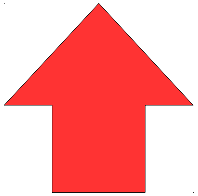
- Architectural concern
- Coupling and Cohesion [Darcy et al, 2005]

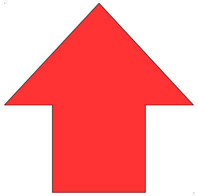
SC definition

$$SC(p) = \frac{\sum_{m \in M(p)} CBO(m) \times LCOM4(m)}{|M(p)|}$$

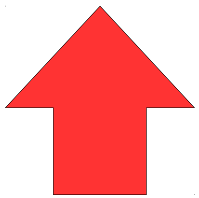
[Chidamber and Kemerer, 1994] (CBO)

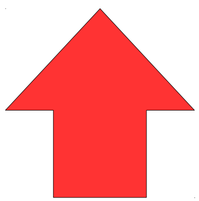
[Hitz and Montazeri, 1995] (LCOM4)

 Structural complexity

 Maintenance effort

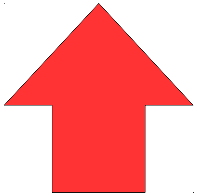
[Darcy et al, 2008]

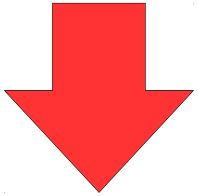
 Structural complexity

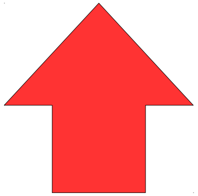
 Maintenance effort

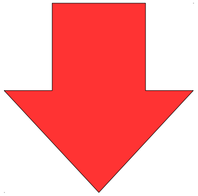
Number of bugs

[Midha, 2008]

 Structural complexity

 Contributions from new developers [Midha, 2008]

 Structural complexity

 Attractiveness

[Meirelles et al, 2010]

(Brazilian Software Engineering Conf.)
Collaboration with ***CCSL - IME/USP*** *(Paulo Meirelles, João Miranda, Carlos Santos Jr., Fabio Kon)*

Research Hypotheses

H_1

changes made by core developers introduce less structural complexity than those made by periphery developers.

$\boldsymbol{H_2}$

among the changes that reduce structural complexity, the ones made by core developers achieve greater structural complexity reduction than those made by periphery developers

Research Design

In this study we **analyse** changes made to the source code of free software projects **for the purpose of** characterization **with respect to** structural complexity added or removed and level of developer participation, **from the perspective of** the researcher **in the context of** the web server application domain.

Research method

Observational study

Unit of analysis

Software change (“commit”, “checkin”)

Independent variable

- L: the level of participation
  - Core
  - Periphery

Dependent variables

- SC: structural complexity
- ΔSC: SC variation in the change
- $|\Delta SC|$: absolute variation

Sample

- Available in Debian GNU/Linux
- Written in C
- Publicly accessible version control repository
- Web server application domain

Data collection

- Version control repository mining
- Determine list of relevant changes (those that actually change source code)
- Extract source metrics and change metadata (author, changed files, date etc)
- Load the data in a relational database for further calculations

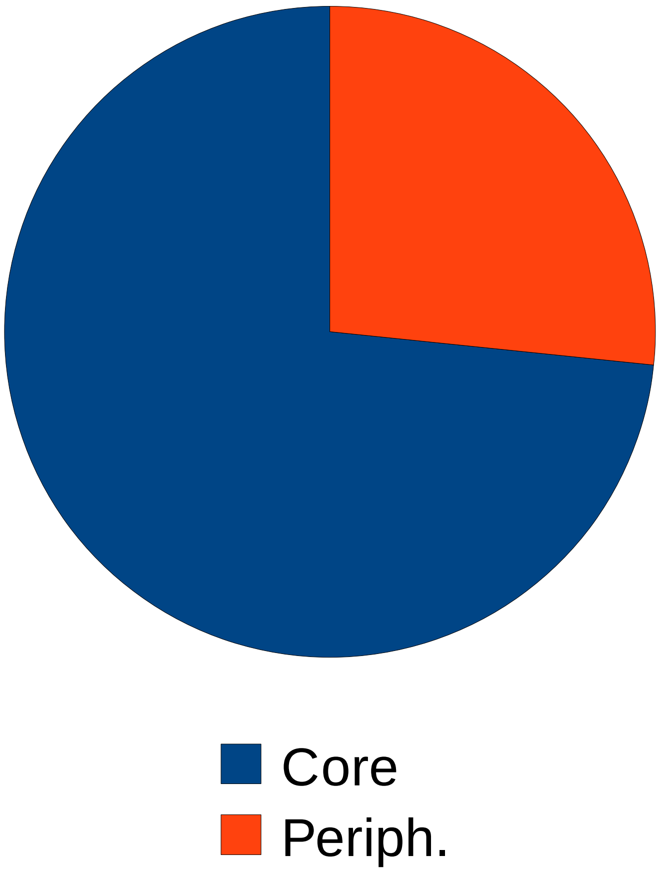
Projects analyzed

Project	Start	End	Commits	Developers
aolserver	2000/05	2009/05	1125	22
apache	1999/06	2009/11	9663	72
cherokee	2005/03	2009/10	1545	8
fnord	2001/08	2007/11	99	2
lighttpd	2005/02	2009/10	775	6
monkeyd	2008/01	2009/06	207	4
weborf	2008/07	2009/10	139	3

Data Analysis and Results

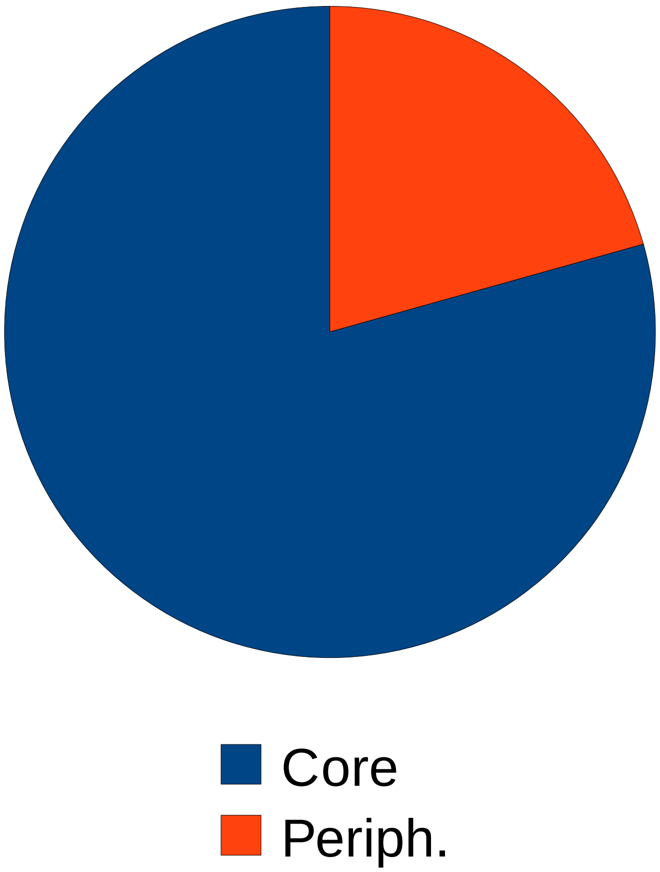
Full dataset

- Available on-line
- 13553 changes
- 9944 by core (73.36%)
- 3609 by periph. (26.63%)



Dataset for testing H_1

- Removed: $\Delta SC = 0$
- 2513 changes
- 1994 by core (79.35%)
- 519 by periph. (20.65%)



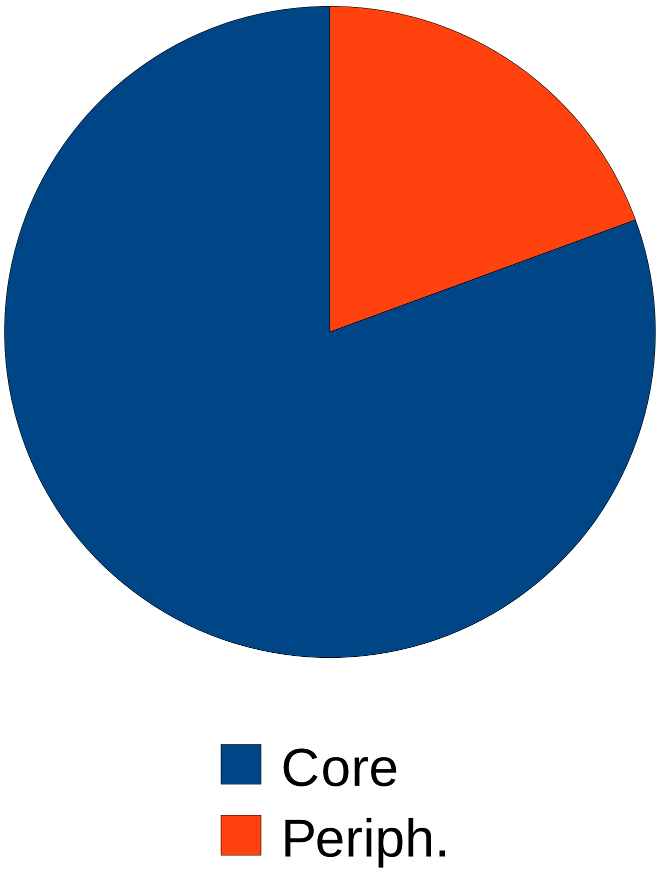
Test for H_1

$$H_1 : \mu_{\Delta SC_{core}} < \mu_{\Delta SC_{periphery}}$$

- Performed a t-test
- Supported by the dataset ($p < 0.05$)
- Changes made by core developers introduce less structural complexity than changes made by peripheral developers.

Dataset for testing H_2

- Kept: $\Delta SC < 0$
- 1165 changes
- 939 by core (80.60%)
- 226 by periph. (19.40%)



Test for H_2

$$H_2 : \mu_{|\Delta SC|_{core}} > \mu_{|\Delta SC|_{periphery}}$$

- Performed a t-test.
- Supported by the dataset ($p < 0.05$)
- among the changes that reduce structural complexity, the ones made by core developers achieve greater structural complexity reduction than those made by periphery developers.

Threats to Validity

On the suitability of the t-test for non-normal distributions

- Sample is “large enough” [Wohlin et al, 2000]
- Wilcoxon/Mann-Whitney test provided similar results

Choice of sample

Sample may not be representative of the population (only C projects)

Single independent variable

Other factors that may affect the structural complexity were not considered

Committers X authors

- Developer identification may be flawed
- OTOH committer participates in the design decision

Nature of changes not analyzed

Type of maintenance may be the actual cause of variation in SC

Conclusions

Difference between core and periphery

Core and periphery provide code of different complexity.

Importance of core team

Responsible for keeping the project's conceptual integrity [Brooks, 1995]

But projects cannot refuse new developers

- Not all projects can keep the same core team forever
- Project management could help new developers

Future work (1/2)

- Testing different developer characteristics
- Testing projects individually
- Richer characterization of changes

Future work (2/2)

- Extending the dataset (app. domains, languages)
- Analyze developer evolution

Thank you

References (1/2)

D. L. Parnas, "Software aging," in ICSE '94: Proceedings of the 16th international conference on Software engineering. Los Alamitos, CA, USA: IEEE Computer Society Press, 1994, pp. 279–287.

K. Crowston and J. Howison, "The Social Structure of Free and Open Source Software Development," First Monday, vol. 10, no. 2, 2005.

D. P. Darcy, C. F. Kemerer, S. A. Slaughter, and J. E. Tomayko, "The Structural Complexity of Software: An Experimental Test," IEEE Transactions on Software Engineering, vol. 31, no. 11, pp. 982–995, Nov. 2005.

S. Chidamber and C. Kemerer, "A metrics Suite for Object Oriented Design," IEEE Trans. Sftware Eng., vol. 20, no. 8, pp. 476–493, 1994.

M. Hitz and B. Montazeri, "Measuring coupling and cohesion in object-oriented systems," in Proceedings of the International. Symposium on Applied Corporate Computing, 1995.

References (2/2)

V. Midha, "Does Complexity Matter? The Impact of Change in Structural Complexity on Software Maintenance and New Developers' Contributions in Open Source Software," in ICIS 2008 Proceedings, 2008.

Paulo Meirelles, Carlos Santos Jr., João Miranda, Fabio Kon, Antonio Terceiro, Christina Chavez. A Study of the Relationships between Source Code Metrics and Attractiveness in Free Software Projects, 2010 (SBES 2010)

C. Wohlin, P. Runeson, M. Host, C. Ohlsson, B. Regnell, and A. Wessl´ n, Experimentation in Software Engineering: an Introduction. Ed. Kluver Academic Publishers, 2000.

F. P. Brooks.Jr, The Mythical Man Month: Essays on Software Engineering. Addison-Wesley , April 1995, ch. "Aristocracy, Democracy, and System Design".