

Revista Eletrônica de Sistemas de Informação

ISSN 1677-3071

v. 13, n. 2

May-Ago 2014 - special issue on FOSS

doi:10.5329/RESI.2014.1302

Editor's space

[Editorial](#)

Carlos Denner dos Santos Jr.

Essays

[RIO IN THE GLOBAL WORLD OF SOFTWARE](#)

Yuri Takhteyev

Free software

[THE ALGORITHMIC AUTOREGULATION SOFTWARE DEVELOPMENT METHODOLOGY](#)

Renato Fabbri, Ricardo Fabbri, Vilson Vieira, Daniel Penalva, Danilo Shiga, Marcos Mendonça, Alexandre Negrão, Lucas Zambianchi, Gabriela Salvador Thumé

[FRENCH SOFTWARE POLITICS: ACTIVISM AND THE DYNAMICS OF GLOBALIZATION FROM BELOW](#)

Pierre-Amiel Giraud, Sara Schoonmaker

[O USO DO SOFTWARE LIVRE NA NA APRENDIZAGEM COLABORATIVA: LIMITES E POSSIBILIDADES DO PROGRAMA "UM COMPUTADOR POR ALUNO"](#)

Rafaela da Silva Melo, Ana Beatriz Gomes Pimenta de Carvalho

[HISTORICAL ANALYSIS OF MESSAGE CONTENTS TO RECOMMEND ISSUES TO OPEN SOURCE SOFTWARE CONTRIBUTORS](#)

Igor Fabio Steinmacher, Igor S Wiese, Andre Luis Schwerz, Rafael Liberato Roberto, João Eduardo Ferreira, Marco Aurélio Gerosa

[SENTIMENT ANALYSIS OF FREE/OPEN SOURCE DEVELOPERS: PRELIMINARY FINDINGS FROM A CASE STUDY](#)

Athanasios-Ilias Rousinopoulos, Gregorio Robles, Jesús M. González-Barahona

[ESTUDIO PRELIMINAR DE LAS EMPRESAS QUE MÁS CONTRIBUYEN AL DESARROLLO DE ANDROID](#)

Sergio Raul Montes León, José David Ramos Benalcazar, Hernán Montes León



This work is licensed under a [Creative Commons Attribution 3.0 License](#).

Esta revista é (e sempre foi) eletrônica para ajudar a proteger o meio ambiente, mas, caso deseje imprimir esse artigo, saiba que ele foi editorado com uma fonte mais ecológica, a *Eco Sans*, que gasta menos tinta.

This journal is (and has always been) electronic in order to be more environmentally friendly. Now, it is desktop edited in a single column to be easier to read on the screen. However, if you wish to print this paper, be aware that it uses Eco Sans, a printing font that reduces the amount of required ink.

RIO IN THE GLOBAL WORLD OF SOFTWARE

RIO NO MUNDO GLOBAL DO SOFTWARE

(paper submitted in September 2013)

Yuri Takhteyev

Assistant Professor (status-only) in the Faculty of Information
at the University of Toronto and Chief Technology Officer at rangle.io
yuri@takhteyev.org

ABSTRACT

While my book (Coding Places: Software Practice in a South American City) dedicates many pages to Lua, it is not about Lua and certainly not about a "syndrome" from which Brazil needs to free itself. The book is about globalization and the paradoxical position cities such as Rio have in the global world of software. One part of that paradox is that Rio is a city like many others. It's not "strange" - it's quite normal. Silicon Valley is unusual - and quite lucky. People working in Silicon Valley do not have to make difficult choices that people in Rio (and many other places) need to make. The book analyses those choices and aims to show how those decisions help maintain the existing order, even as they tug on it from the edges. It is not my intention to condemn those choices, however. I would probably make the same choices myself. In fact, I often do (My own book is also only available in English). Yet I think it's worth reflecting on those choices more, considering them in their complexity.

Key-words: software development; legitimacy; place.

RESUMO

Embora meu livro *Coding Places: Software Practice in a South American City* (Lugares de desenvolvimento: a prática de software em uma cidade sul americana, em tradução livre), a que se refere este ensaio, dedique muitas páginas à linguagem Lua, ele não é sobre Lua e, seguramente, não é sobre uma "síndrome" da qual o Brasil precise se livrar. O livro é sobre globalização e a posição paradoxal de algumas cidades como o Rio de Janeiro no mundo global do software. Uma parte do paradoxo é que o Rio é uma cidade como muitas outras. Não é "estranho" – é absolutamente normal. O Vale do Silício é incomum – e muito afortunado. As pessoas que trabalham no Vale do Silício não precisam fazer as escolhas difíceis de quem mora no Rio (ou em muitas outras partes). O livro analisa essas escolhas e procura mostrar que elas ajudam a manter o *status quo*. Contudo, não é minha intenção condenar essas escolhas. Eu provavelmente também as realizaria. Na verdade, o faço com frequência (meu próprio livro está disponível apenas em inglês, não havendo uma edição em russo). Ainda assim, acho que vale a pena refletirmos sobre essas escolhas, considerando-as em toda a sua complexidade.

Palavras-chave: desenvolvimento de software; legitimação; lugar.

1 INTRODUCTION

In June 2005 I arrived in Rio de Janeiro on a flight from San Francisco to interview Carioca software developers about their work. I was doing my Ph.D. at Berkeley, looking at software development as a case of globalized modern occupation. I had by that point lived in San Francisco Bay Area for many years and had worked for some time in the area known around the world as “the Silicon Valley.” (The locals usually say “South Bay” when talking about the actual place, and “the Valley” when talking about the tech phenomenon). Wanting to now try something different, I decided to focus my research on some other place. I eventually picked Rio. Over the course of the next several years I spent many months in Rio interviewing software developers and sometimes working side-by-side with them on their projects. The project resulted in a book, published in 2012 by the MIT Press, entitled *Coding Places: Software Practice in a South American City*.

But why study software development in Rio? What is so special about it? I heard those two questions many times over the course of the following years, from Americans and Brazilians alike. The answer to the second question is, in a way: “Nothing really”. Rio de Janeiro is one of the many cities around the world where software developers work. Nothing about their work necessarily stands out relative to the work of people in other, similar, cities. Rio is not a rival to Silicon Valley and unlikely to be in the future. It probably is not even the strongest contender for this title within Brazil.

But then, why write a book about software work in Rio de Janeiro?

I argue in my book – written for the global, and in particular US-based audience – that we have much to learn about software by looking at it in places like Rio de Janeiro, because Rio is, in many ways, much closer to the “typical” place where software work is done. For all the buzz associated with Silicon Valley, *most* software work happens in places that are more like Rio than they are like San Francisco. Which is to say, most software work is done in places that share an interesting characteristic: on the first glance, each of them might seem like an unlikely focus for an ethnography of global software. Yet it is in places like that that the heavy lifting of globalization takes place.

2 RIO'S GLOBAL PERSPECTIVE

Two things stood out for me early in my interviews with Rio's software developers. First was the extent to which software work in Rio was *similar* to software work I had seen elsewhere. As one of my interviewees put it, “a server is a server,” whether you work on San Francisco's Market Street or Rio's Avenida Rio Branco. Rio developers' sometimes explained this similarity as a matter of pragmatism: why reinvent the wheel? But it clearly went beyond that. My interviewees did not just talk about using the

same tools and methods that I had seen in the United States; they also seemed to be making the same jokes along the way. They were eager and competent members in the same global cultural space in which I was myself a member, despite the fact that most of them had never left Rio, the city I was only starting to get to know.

This similarity might seem natural to those who inhabit this shared cultural space, but its oddness can be highlighted through a contrast with music. For all of its internal diversity and complex ties with the global music scene, “Brazilian music” can be easily identified as being distinct from “American” music. In contrast, “Brazilian software” is in many ways a meaningless term. Apart from a few policy makers, none of the people I met in Brazil seemed to be engaged with “Brazilian software”. They were people doing software, who just happened to be in Brazil.

Many of my interviewees saw this similarity of work as natural, just the way software works. However, I soon came to see it as a product of their own work. And this was closely linked with the second thing that stood out for me: Rio developers’ outward orientation. While Rio’s software scene at times looked a bit provincial to me in terms of scale (smaller, less ambitious companies than in San Francisco, fewer events with fewer people), I could hardly accuse it of being parochial. In fact, Rio seemed to offer a broader, more globalized perspective on the world of software, a perspective that was much harder to get in San Francisco. My interviewees were spending a lot of time keeping up with what was happening outside Brazil and stressed the importance of doing so.

In other words, they were eager and competent members in the global software collective because they put much effort towards that end. Or, to put it just a little differently, the practice and culture of software development got to be so seemingly uniform around the world because people in places like Rio put a lot of effort to make it so.

Such efforts to stay in sync, however, also have costs. The most significant one is that they can challenge local alliances. There is also a deep irony in such efforts. Looking outwards to remote places such as Silicon Valley is one of the things that makes Rio *different* from Silicon Valley. After all, Silicon Valley is notorious for its insularity.

3 GLOBAL WORLDS OF PRACTICE

Software has a strange relationship with globalization. On the one hand, pundits professing the “death of distance” in modern world typically point to information technology as the main driving force and often cite software development itself as an example of work that can be done anywhere (CAIRNCROSS, 1997; FRIEDMAN, 2006). Yet we know that place matters in today’s world – and some scholars argue that they matter more than ever (FLORIDA, 2008). Ironically, software development again comes in as a text-book example, being famously associated with Silicon Valley, followed by a number of secondary clusters (SAXENIAN, 1996; ZOOK,

2002). The existence of such clusters reminds us that place does matter – and perhaps more in software than in other fields.

Thinking about this problem theoretically, one finds no shortage of literature explaining why place matters. Some of the reasons for local industry clustering today are the same that Marshall ([1890] 1927) pointed to when analyzing the world of 19th century England. It is easier for workers to find jobs in a place where a lot of employers are located and it is easier for the employers to find workers in places where there are a lot of potential employees. Additionally, some places develop long-term complementary relationships between specific industries and local institutions (SAXENIAN, 2006). Work on Information and Communication Technologies has also shown that the use of communication technologies is often complementary to face-to-face interaction (WELLMAN, 1979, HAMPTON AND WELLMAN, 2003).

This focus on the importance of the local, however, can give us a rather limited view of what is happening in software. It doesn't account for the fact that the practice of software development is, at the end of the day, remarkably similar across the globe.

What I found in my own work was a need for a theoretical counterbalance to the notion of place, a way of thinking about the loose yet real collective entity that united millions of software developers around the world. I have gone through a number of concepts. Should we think of this as “the software industry”? The “software community”? The “network” of software developers? The theoretical chapter of the book deals with the specific problems of those concepts. Overall, however, the main challenge lay in balancing two sides of software work, which correspond to two kinds of analysis that are known in today's social science as “cultural” and “politic-economic.”

On the one hand, being a software developer is about identity. It involves subscribing to certain cultural norms and adopting a certain way of seeing the world. It also involves a sense of belonging in the collective of people engaged in the same work. For many people, this sense of belonging is so strong that they even believe that you must have a certain type of brain to be a software developer. If that was the case, of course, software development would be something you would need to be born in – akin to a tribe.

Yet at the same time, for most people who do software, it is a job – a job that comes with a paycheck on which they rely for paying their bills. This paycheck becomes possible through the existence of a larger system of economic transactions. Consequently, to understand software work we must look at how it is embedded in this larger system.

Reconciling those two sides can be difficult. Most theoretical accounts fall on one side or the other. The “cultural” side of software, for example, can be easily understood through the work of the scholars in the Chicago School of Sociology (e.g., HUGHES, 1958; BECKER, 1963). Their work,

however, has often been accused of politic-economic naivete by scholars working in the more recent Marxist tradition (in particular those following, Braverman, 1974). The latter approach, however, is often dismissive of the workers' cultural commitments to their work, often treating them as self-deception (e.g., BURAWOY, 1979).

I develop an idea of software as "a world of practice" to counteract the limitation of the two approaches mentioned above. One key component of "worlds of practice" is the notion of practice. A complex concept with rich and sinuous history, the modern use the notion of practice in social science is perhaps best summed up with a definition provided by Schatski (1996): "a temporally unfolding and spatially dispersed nexus of doings and sayings" (p. 89). A practice is a system of conduct that maintains continuity in time and place. The mechanisms though which this happens involves people recognizing patterns in social conduct and following those patterns because doing so helps them achieve their goals (GIDDENS, 1979).

The second key component is the notion of practice as forming a "world" – a bounded system characterized by shared understandings and joint projects. Boundaries of such worlds divide the universe into members and non-members. This means, among other things, that an individual must often establish him or herself as a member in the world before gaining access to its material resources.

For some worlds, the boundaries can be quite formal. In others, like the world of software development, they are much more subtle. This, however, does not mean that they are easier to cross. There is no organization that one can join to definitively be accepted as a *bona fide* software developer.

Such processes of authentication create challenges in places at the periphery of the world of software. Nobody would wonder if an engineer working for Google in Silicon Valley is a "real programmer" or is just faking it. My interviewees in Rio, however, seemed to run into such questions quite often. Were they, individually and collectively, software developers in the same sense as their counterparts in Silicon Valley?

Perhaps the most interesting aspect of this process is that it is rarely the case of software developers' in Rio having their credentials checked by those in Silicon Valley. What matters most, practically speaking, is the local judgment, in particular that of clients and employers. The clients want "real" developers, of the global kind, not local surrogates. (Part of the reason for that is that the clients are themselves pursuing their own globalization projects.) To convince the client, a software firm must be seen as global by the local community. For that, it must hire developers that would support such global aspirations.

The best way to achieve globalized recognition locally is to actually have ones work recognized abroad. For those who cannot meet this

standard, displaying a global perspective becomes the next best option. This, in turn, often means shunning anything that may seem parochial.

4 THE LOCAL COSTS OF GLOBALIZATION

The resulting dynamic has important benefits. It keeps out parochialism and helps bring to a place like Rio de Janeiro the best remote practices. Keeping up with such practices becomes everyone's job!

Yet it also has downsides. Most importantly, it strains local alliances. Local, face-to-face interaction has many benefits, as was mentioned earlier. It turns out then, that people working in a place like Silicon Valley can have it both ways: they can collaborate locally, reaping the benefit of face-to-face contact, while at the same time maintaining a pristine global brand, since the local practices of the place where they work serve as a yardstick for what counts as "global" elsewhere. Those working in a place like Rio often face a choice. They can focus on building local alliances – and run the risk of being seen as parochial. Or they can build global links and bear the cost of doing things over long distance. In practice, they usually try to find some balance. The book explores several different ways such balancing act can work. Here, however, I will only briefly talk about one case that I studied closely: Lua, a programming language developed at PUC-Rio that has achieved remarkable success around the world.

When I first encountered Lua in 2005, nothing about it struck me as notable: a programming language developed in a university, of which there are probably thousands. In particular, I did not even perceive much of a buzz about Lua in Rio. People I asked about either had not heard of it, or described it as just some project by "the guys at PUC", without providing much of an endorsement.

My opinion of Lua changed a year later, in 2006, when I saw a discussion of Lua on Slashdot, while back at Berkeley. A number of commenters professed their love for Lua, which they described as a beautiful language, "light-years ahead of Ruby." Yet more interestingly, they pointed to the many applications built with Lua, in particular World of Warcraft, a game that was then on its way to a Guinness World Record award as the most popular multiplayer online game.

When I decided to interview some Lua users in San Francisco Bay Area, I had no trouble finding people to talk to. All of them told me Lua was a great language, combining the semantics of Scheme and Self with accessible Pascal-like syntax. They often compared Lua to JavaScript, pointing out that Lua was, in many ways, "JavaScript done right." Like JavaScript, Lua was primarily used as an embeddable scripting language. Unlike JavaScript, it was small and elegant.

When I returned to Rio in 2007 I decided to dedicate much time to studying Lua. I initially thought that this would involve talking to software developers at companies using Lua. There weren't many to choose from. My four month investigation of the Lua scene in Rio covered most people

who had much to say about Lua. Yet for the most part it involved shuttling between PUC-Rio and the one company in Rio de Janeiro that had bet its fortunes on Lua. It became clear to me, that San Francisco was a better place to look for Lua programmers than Rio de Janeiro. (Lua's later adoption as a part of the SBTVD digital standard promised to change this, but seems to have had a lot less effect that one could have expected).

To explain the discrepancy between Lua's local and global adoption I look at the commitment to global ties exhibited both by the Lua's team and by the potential adopters in Brazil. The starkest example on one side is Lua's commitment to English. Numerous books about Lua have been written in English over the last ten years, including three editions of *Programming in Lua*, a book by one of Lua's authors. A proper Portuguese book on Lua is yet to come out.

The reverse is very much true as well, however. Until Lua achieved popularity abroad, it seemed to present little interest to Brazilian software developers. For some, it was a case of "only in Brazil and not Jaboticaba." For others, it was a matter of pragmatism: no one gets fired for choosing a US-built solution over a local alternative. This situation has changed somewhat as Lua's global popularity grew in the recent years. By this point, however, Lua had spent years being shaped by the needs of its foreign users. Consequently, it may not be the best choice for software work in Brazil.

This analysis is not intended as a condemnation of Lua's approach. Lua did not find much use in Rio de Janeiro, but it did help bring the city some recognition as a place where good software is built. Plus, a strategy of focusing closely on local adoption would have likely led to less success. Lua also managed to achieve substantial success without support of a large local community, so perhaps it did not need it, though its current configuration implies the difficult challenge of serving an community of users who are always far away. What this analysis highlights is the need to make choices – choices that developers located in the "centers" of the world of software do not need to make.

REFERENCES

- BECKER, Howard S. 1963. *Outsiders: studies in the sociology of deviance*. New York: Macmillan.
- BRAVERMAN, H. 1974. *Labor and monopoly capital*. London: Monthly Review Press.
- BURAWOY, M. 1979. *Manufacturing consent*. Chicago: University of Chicago Press.
- CAIRNCROSS, F. 1997. *The death of distance: how the communications revolution is changing our lives*. Cambridge, MA: Harvard Business School Press.
- FLORIDA, R. 2008. *Who is your city?* New York: Basic Books.

- FRIEDMAN, T. L. 2006. *The world is flat: a brief history of the twenty-first century*. New York: Farrar, Straus and Giroux.
- GIDDENS, A. 1979. *Central problems in social theory*. London: Macmillan.
- HAMPTON, K.; WELLMAN, B. 2003. Neighboring in Netville: how the Internet supports community and social capital in a wired suburb. *City and Community*. v. 2, n. 3 (Fall), p. 277-311. <http://dx.doi.org/10.1046/j.1535-6841.2003.00057.x>
- HUGHES, E. 1958. *Men and their work*. Glencoe, IL: Free Press.
- MARSHALL, A. [1890] 1927. Book IV: The agents of production. Chapter X. Industrial organization, continued. The concentration of specialized industries in particular localities. In: *Principles of economics: an introductory volume*, p. 267–277. London: Macmillan.
- SAXENIAN, A. 1996. *Regional advantage: culture and competition in Silicon Valley and Route 128*. Cambridge, MA: Harvard University Press.
- SAXENIAN, A. 2006. *The new Argonauts: regional advantage in a global economy*. Cambridge, MA: Harvard University Press.
- SCHATZKI, T. 1996. *Social practices: a Wittgensteinian approach to human activity and the social*. Cambridge, UK: Cambridge University Press. <http://dx.doi.org/10.1017/CBO9780511527470>
- WELLMAN, B. 1979. The community question: the intimate networks of East Yorkers. *American Journal of Sociology*, n. 84, p. 1201-31. <http://dx.doi.org/10.1086/226906>
- ZOOK, M. A. 2002. Grounded capital: venture financing and the geography of the Internet industry, 1994–2000. *Journal of Economic Geography*, n. 2, p. 151-177. <http://dx.doi.org/10.1093/jeg/2.2.151>