

Jeeks: Developers at the Periphery of the Software World

Yuri Takhteyev
UC Berkeley School of Information

The paper looks at the experience of software professionals in Rio de Janeiro, Brazil and their forms of engagement with the larger, global world of software development. The developers, mostly young white men educated in local universities, spend their days in office buildings of downtown Rio de Janeiro writing software code to support the information technology needs of local organizations. Most of them have lived their whole lives in Rio de Janeiro and have no personal contacts abroad. From their offices, however, they regularly access technical documents authored by software developers half the way across the globe, using such documents both to keep up with the rapidly evolving foreign technology from which they build their local systems and to find solutions to specific problems that they face.

Engagement with such documents also helps them strengthen their claim to membership in a profession that is global even in its local name - “desenvolvedor de *software*,” a claim further strengthened by extensive adoption of foreign software culture. The work of Rio software developers thus involves daily contact and identification with “global” practices, global culture and global discourses, while at the same time being crucially linked with local networks, making their practice *simultaneously* local and global. Moreover, the global is not merely a larger container for the local: rather than just participating in the global practice through their “local chapter,” the developers are involved individually and *directly* in the local and global networks.

While developers often talk quite casually about their use of global technology, their

words betray the difficulty of such dual participation. The recognition of the technology they use as *foreign*, not theirs, and the incompatibilities between the globally sanctioned practices and local realities lead to deep tensions between their local, national and global commitments. The paper uses the data from a series of qualitative interviews with Rio software developers to explore those tensions, looking at the experience of Brazilian developers as a case of peripheral membership in a global and increasingly integrated professional world.

The Software World

Qualitative studies of work have typically looked at workers as embedded either in specific organizations or in “communities” that cross organizational boundaries. Within the latter approach, Van Maanen & Barley (1984) introduce a notion of an “occupational community” as “a group of people who consider themselves to be engaged in the same sort of work; whose identity is drawn from the work; who share with one another a set of values, norms and perspectives that apply to but extend beyond work related matters; and whose social relationships meld work and leisure” (p. 287). A closely related notion of “communities of practice” (Lave & Wenger 1991) similarly stresses identity and shared meaning. In their classic form, both occupational communities and communities of practice are understood to be small and highly localized, as is, for instance, the case with Orr’s (1996) celebrated ethnography of Xerox reps.¹

While both “occupational communities” and “communities of practice” do allow for the possibility of distributed communities, such communities are typically expected to be tightly-

¹ Van Maanen & Barley (1984) in fact argue for the need to look for smaller units than may be obvious at the first sight, identifying, for example, several distinct occupational communities among fishermen of Gloucester, MA.

knit. For example, Van Maanen & Barley use American symbolic interactionists as an example of a geographically dispersed occupational community, while stressing that “policemen” in general should be seen not as a single community but rather as a collection of many local ones. Communities of physicists famously described by Traweek (1988/1992) and Knorr Cetina (1999) are also examples of such distributed yet tightly-knit communities.

If we understand “occupational communities” as local, the software developers of Rio de Janeiro are involved in a number of local occupational communities. Most of them have spent all of their lives in the same city, though they move frequently between companies within it (as is the case in California – see Saxenian 1996). They typically have friends among other developers and interact with each other in ways that “meld work and leisure” (Van Maanen & Barley 1984, p. 287), especially within smaller communities that unite developers working with similar technology (e.g., developers using the Java programming language or those working with mobile phones).²

Their experience and their understanding of their occupation, however, cannot be understood through a narrow focus on such local interaction, since they see themselves as first of all “software developers” (or “nerds” or, more narrowly, “Java developers”, “Perl developers,”

2 The following two quotes describing discussion of technology between colleagues/friends can illustrate such melding. One interviewee talks about discussing technology in general over billiards:

Claudio: We were playing [billiards], and started chatting about technology, talked about a digital camera - how many megapixels. And from there to file compression technology, then went on to cell phones with cameras, started talking about cell phone technology. Talked about GSM, CDMA, CDMA [unclear]. From there got to - those technologies that are coming up with cell phones. And we weren't trying to do anything with this – we were just playing billiards.

Another developer talks more specifically about asking a friend for technical advise on something that he “wrecked his brains” doing:

Jaime: [S]ometimes we got get lunch, chat informally about I-don't-know-what, one chats, one talks about the family, another about his wife, then the conversation turns to “Look, the other day wrecked my brain doing such and such routine. Do you have any idea how to do it?” So the guy goes: “Yeah, already did this routine, use such-and-such library.” “Ah, cool. Great.” So, I went back home – yay, it worked!

“PHP developers”) in a global sense of the word. This global sense can only be understood through a reference to the larger world of software development.

My Brazilian interviewees display this global identification, for example, when ask they me whether I am a software developer myself. When they do this, they are not inquiring whether I am a member of the local occupational community (which they usually know I am not), but rather whether I belong to a broader category of people who write software code and with which they themselves associate. As a foreign member of this group, I am not expected to understand *local* meanings and norms. For instance, the developers take time to explain to me the many difficulties of doing business in Brazil. At the same time, they expected to understand their technical jargon (“function,” “library,” “compiler”), as well as certain values (understanding what constitutes “good code” and why Linux a better operating system than than Microsoft Windows) and practices (knowing what it means to look on the Internet for a library.) Many of them also expect me to have some familiarity with the broader “nerd” culture, of which they consider themselves a part. This broader cultural knowledge would include, for example, familiarity with a wide range of computer games and comic book heroes. Being identified as a former software developer I in fact need to make special effort to make my Brazilian interviewees *suspend* the assumption that I share their meanings and opinions and to explain to me what they mean, for instance, by “looking” for code on the Internet, *as if I was not one of them*.

When talking about this larger world of software developers to which both I and my Brazilian interviewees belong, it may be tempting to refer to it as the global “community” of software developers. Such usage would be misleading. Brazilian software developers differ in important ways from Van Maanen & Barley’s symbolic interactionists or Traweek’s and Knorr

Cetina's physicists, since most of them have never personally interacted with software developers outside Brazil. Instead, their daily involvement in the global practice is mediated through texts and artifacts. They are part of the global software technoscience network understood in actor-network sense (e.g., Latour 1988), but are rarely a part of global networks in the "social networks" sense.³

This looser collectivity of software developers world-wide can be better understood using the concept of a "social world" (Shibutani 1955; Strauss 1978, 1984; Unruh 1980), which allows for a discussion of geographically dispersed collectivities of people without the connotation of close interactions.⁴ The social world of software development (or "software world") is truly global in its spread, yet has a clear geographic center on the West Coast of the United States. Within that general area, two specific places serve as a source of tremendous symbolic power. One is the Silicon Valley - the area between Palo Alto and San Jose, about 40 miles south of San Francisco, which is home to the Google campus, the single most important place in the imagination of the developers.⁵ Another one is Redmond, WA – the home to Microsoft. While the geographic distance between those two places is quite visible to developers in the United States, it is somewhat blurred when looking from Rio de Janeiro - both places become "there in the United States" (*lá nos Estados Unidos*).⁶

Borrowing a term from Levine's (1972) study of the Chicago art world, we can describe

3 Their experience is thus very different from the "argonauts" described by Saxenian (2006) or Castells' (2000) "networkers" who "shift and commute between nodes of the global networks that control the planet."

4 An alternative model for talking about the global "software world" is the concept of "networks of practice" discussed by Brown & Duguid (2000) and Duguid (2005).

5 For a discussion of Google in the imagination of Brazilian software developers see page 3 in Takhteyev (2007).

6 This geographic blurring does not, however, imply blindness to the difference between the two centers. Much like many of their counterparts in the United States, Rio developers often see those poles as representing, respectively, the poles of "good" and "evil" in the software world.

those centers as the “Meccas” of the software world – places that possesses tremendous symbolic power and serve as a source of legitimation and arbiters of “success” for local practices. Local processes occurring away from the centers invariably orient themselves in relation to such Meccas and it is through their power that the software world stays together (much like the real Mecca remotely holds together the Muslim *umma*).

This rest of this paper explores the interactions between the local occupational communities formed by Rio software developers and the larger software world of which they strive to be a part. I first look at the imperative to engage *directly* with foreign sources of knowledge, and then at the tensions and contradictions this imperative creates, given the developers inability to altogether ignore their geographic location.

This Study

This paper is based primarily on a series of interviews with software developers conducted in Rio de Janeiro in the summer and fall of 2005. 50 software professionals were interviewed in this round, most of whom worked as software developer with a few others doing academic research in computer science or managing software development. The participants were recruited through a snowball sample with an effort to maximize diversity of work environments (public vs. private companies, large vs. small) and educational levels (from high school to Ph.D., with most developers having a bachelor’s degree or working towards one). The interviews lasted from 40 minutes to 3 hours and were conducted either in English or Portuguese depending on interviewees preferences and level of English proficiency.

In 2007 (March – August) I conducted another round of fieldwork that included

additional interviews and participant observation of two software projects. While the data collected from this second round allow for a more nuanced analysis of some of the issues discussed in this paper, exploring the diversity within the local communities, it does not appear to contradict the main points of this paper.

“Directly to the Source”

Much like in many other places (including the United States),⁷ the work of software developers in Rio de Janeiro is characterized by regular use of technical documents as a part of daily “research” (*pesquisa*) – task-specific learning that developers must undertake when solving specific problems. For example, when asked what he does at work a developer says:

Flavio: Most of the time I am developing. But to develop is not just to write code, right? There is also a large part of searching, studying, looking at specific topics, finding answers to questions, looking for references. There is also this set of things.⁸

Such research often initiates from a new requirement from the management (“add the ability to generate PDF reports,” “demonstrate to the client that the new server is faster”) or an unexpected problem, which present the developer with a task they have not faced before. Alternatively, the developer may be alerted to a new technology in a conversation or while looking for something else and decide to research it to see if it may be useful for current or future tasks. While interactions with local friends and colleagues play an important role in this process of research (see Takhteyev 2005), the developers nearly unanimously stress the overwhelming importance of

7 The comparison’s with the United States made in this paper are based on the author’s personal experience working for three years as a software developer and a manager in Silicon Valley as well as on available literature on software development in the United States.

8 See <http://takhteyev.org/brazil2005/quotes/asa-2007/> for the original Portuguese text of the quotes used in this paper.

the Internet, often sparing no colorful words.⁹

The developers stress the pragmatic nature of their use of the Internet, pointing out that a Google search often yields results in less time than it would take to walk over to ask a colleague. The integral role that research plays in developers' work also means that asking colleagues for things that one could find on the Internet may be seen as asking them to do one's work.¹⁰ Use of such resources, however, also plays an important symbolic function, as it highlights the developers' links to the larger world of software development. Through their use of the Internet, the developers not only solve the practical problems that they face but also assert – individually and collectively – their membership in the global software world and their right to be called “software developers” (*desenvolvedores de software*) in a *global* sense rather than being seen as involved in an idiosyncratic and local practice on the wrong side of the “idea gap” (Romer 1993).

The resources used in the course of daily “research” are usually foreign. The developers rarely mention the language of those resources, but when asked explicitly, they commonly say that almost all resources that they use (online texts as well as printed books) are in English. The use of English is often discussed as non-remarkable, something that developers *obviously* do.

9 For example, the developers talk about the Internet as “the worlds greatest library” where one can find “all imaginable and unimaginable resources.” Older developers often also specifically highlight the change that the Internet has brought, contrasting it to their earlier experience. For example, comparing his current experience to mid 1980s, one of the interviewees says:

Vinicius: Then if you knew that the person knew about it, you would spend more time trying to talk to him. It's not necessary anymore. You don't need to, actually... And again, this is primarily due to the Internet. You can get any kind of information you want on the Internet.

10 The idea that a software developer must know how to *resolve* problems rather than ask questions can be illustrated by the following quote:

Walter: In my first internship my boss told me something that I remember until today. “I don't pay you to bring me problems. I pay you to solve my problems.” And then what happened: from that day until now I have only been solving problems.

In other words, a software developer is paid to find answers to technical questions and passing those questions to others is subject to careful accounting - “You don't want to be the person who is always asking questions.”

When asked why they use English the interviewees typically stress the limited availability and the low quality of the resources in Portuguese, arguing that too much is “lost in translation” (*tradução perde*) in case of books and that there simply too little technical materials in Portuguese on the web.

The stress on using English, however, quickly betrays the tremendous symbolic power associated with the language and interviewees’ fears of being seen as sub-par software developers for lack of English skills. The idea that good developers can and do read English is often expressed quite strongly:

Mario: And the lists in Portuguese, they are for... Maybe this is my prejudice, but normally those lists have more people who are there because they don't know English. Because this is the only resource that they have.

Mario continues to explain that his company recently mistakenly hired a developer who could not read English. “That’s terrible for someone who is supposed to work in computer science,” he explains. (The company since introduced an English test for all new job candidates.)

At least some Brazilian developers quite obviously do participate in discussions on Portuguese mailing lists and online forums and use technical books in Portuguese . The bookstores are full of them and one can sometimes see translated books on developers’ desks. There is usually a ready explanation, however: “A friend gave it to me” or “I didn’t have time to order it on Amazon so I had to get it downstairs.” The same reluctance to admit use of local resources or any difficulties with English also comes up frequently in interviews, especially with the better educated developers:

Yuri: You always use words in English [to search]?

Edmundo: In English. Always. I beg your pardon: 80% in English, 20% in Portuguese. And I will tell you why. There is a strong Java community here in Brazil. [...]

Yuri: And how do you choose [which language to use]?

Edmundo: How do I choose... Good question. Do you *really* want to know? When I am tired to write in English, then I enter it in Portuguese. When I am tired to write in English. [*Embarrassed chuckle.*] There isn't a pattern. When I don't want to read English, when I don't want to read *anything*.

Yuri: So, reading English is more difficult for you?

Edmundo: Because it's not my native language. It's not my native language. Like you, even though you've lived in the United States already for six years, or three years, or was it six?

Yuri: Ten.

Edmundo: Ten? Ten years, damn! And still it is easier for you to speak Russian.

What is significant about this passage is not the fact that Edmundo finds it more difficult to read in English than in his native language, but his reluctance in admitting that. After first telling me that he uses English “always,” he then justifies occasional use of Portuguese by the relative strength of the Java community in Brazil, then finally says it is a matter of being tired. Edmundo sees his use of Portuguese as a weakness – a weakness that he hopes I would understand given my own status as a non-native English speaker.¹¹ As most interviewees he feels that a software developer is *supposed* to stay up-to-date (*se manter atualizado*) in the world of technology by going “directly to the source” and using the dominant language of that world rather than getting the information second hand.¹²

11 In the larger context of the conversation it is quite clear that the statement that the local Java community is “strong” must be read as “not as weak as you might think.” Most developers approached the interviews with an assumption that my research aimed to compare them negatively with the American developers. On the other hand, my status as a non-native English speaker, often eased those tensions and opened many doors. In many cases the interviewees talked a lot more frankly after getting to chance to ask me about my own background and eventually started mentioning the fact that I was born in Russia as a part of the introduction.

12 Of course, the pressure to stay “up-to-date” is hardly unique to Brazilian developers. Barley & Kunda (2004) describe in some detail the same concern among American IT contractors, even though they attribute it

The desire to stay up-to-date and do so through English must be understood in the context of developers' work to assert – both individually and collectively – their membership in the global software world revolving around distant Meccas. Collectively, such claim to membership requires an assertion that local software development work is no different from that done in places like Silicon Valley. Individually, it requires that developers have *direct* connection to the foreign source of technological knowledge, unmediated by local people and resources.

When asked about the difference between software work in Brazil and other places, Brazilian developers often stress that their work is no different. “A server is a server” they say, whether it is running in Rio de Janeiro or Palo Alto, and the same knowledge is required to set it up in either place. While this statement largely *is* true, it is important to realize how much rests on it for Brazilian developers: if indeed “a server is a server,” then they are “software developers” with no further qualifications and their work and their claim to membership in the global software development community is no different from that of American developers. If software work in Brazil inherently depends on “local knowledge” and so does local work in California, then they are *Brazilian* software developers – a world apart from their American counterparts.

It would thus be incorrect to either assume that their work is highly idiosyncratic and entangled in local knowledge or to think of it as naturally universal. Instead, the universality of software knowledge is very much constructed through daily local work. Such work may or may not be done with an intention of re-enforcing the universality of software knowledge. Even when the developers search for foreign solutions and engage in foreign documents simply as a way of

specifically to *contractors*, who they contrast explicitly with full-time engineers in Kunda (1992). However, American IT workers do not face the choice between the “local” and the “global” language when trying to track technical change. The peripherality also adds the pressure of staying up to date *collectively* and the tensions between collective and individual advancement.

solving a problem at hand, they are engaged in work which helps make foreign knowledge increasingly applicable in the local context. Quite often, however, the developers seem to be aware of the relationship between their use of foreign resources and their collective claim to being software developers.¹³

Local competition also creates an imperative to demonstrate that they are up-to-date *individually* or as companies. Many developers talk about the need to not be caught off-guard by a question about new technology they might not be familiar with. A manager of a small company, who describes himself as “an engineer”, says:

Sergio: We have to go, for example, to a company to show our product and the guy asks us about open source software. Another open source software that makes things related to what we are doing. If I don't know the open source product...

Yuri: Something recent?

Sergio: Yeah, always the guys in the company – they read... For example, “Oh, your software works with Hibernate.¹⁴ I read in a newsletter that Hibernate has this or those problems or these or those characteristics.” [...] “Isn't Hibernate difficult to implement because of this or that?” Even if you are the market guy or the sales guy you have to know how to discuss about this. So we have to be always keeping in touch with those things. [...] Every time I am selling our product to a company, the business guy calls the tech guy so they can have an assistant to discuss it. And always the tech guy wants to double-cross you. He gets a point in your technology and he tries to understand why they

13 In addition to recognizing the *ongoing* work of making foreign knowledge applicable, it is important to recognize the historical work that had to be done. Developers' ability to claim that “a server is a server” has a lot to do with the current ubiquitous use of foreign hardware and software platforms, which arrived to Brazil only after long resistance (see Tigre 1983, Adler 1987, Evans 1989, Dantas 1988, Evans 1995). Latour's (1988) discussion of science as a heterogeneous network (“technoscience”) provides a framework for understanding this process theoretically. Considering how Pasteur's bacteriology ends up being successful in stopping anthrax on farms (i.e., outside the narrow confines of the laboratory), Latour (1988) suggests that Pasteur had to “transform enough features of the farm into laboratory-like conditions” before bacteriological knowledge could have power on the farm. Similarly, when faced with the seeming ease with which Brazilian developers download and use foreign technical solutions, we must consider the extent to which they have already been “enrolled” into the network of global software technoscience. This process of enrollment must be understood historically, where each round of enrollment sets the stage for the next.

14 Hibernate is a popular open source technology for linking Java software with databases. It's development is currently lead by Red Hat, a US-based Linux vendor. At the time of the interview Hibernate was a popular “new” technology.

cannot use your product, because he wasn't the one who had chosen you.

Yuri: And you are saying that they generally tend to know about all the latest things?

Sergio: Yes, they do. [...] When they get the specifications of our product they try to get some buzzwords or words at the end, and then they find, they read, they do a lot of things. So they get some questions – to show that they are working.

The developers and their companies thus have to constantly one-up each other in the ongoing technology race. This competition takes place not only between the companies but *within* them.

In the quote above, the interviewee refers to the client's "tech guy's" need to show to his boss that he has a better grasp on technology than my interviewee. The standards in this race are always set by foreign technical knowledge and Google is the primary weapon.

Individual strategies vary: some engineers attempt to "surf the technology wave," getting an early start on a new technology, selling their skills at a premium while there is little competition, then moving on to the next thing; others try to minimize the time they spend reading about new technology and to get by with what they know for as long as they can. Even such "late adopters" (Rogers 1995), however, talk about the pressure to stay up-to-date and to do so through direct engagement with foreign sources.

Membership in the software world achieved through similarity of practice is further fortified by the borrowing of American software *culture*. Brazilian developers display familiarity with the jargon, lore and values of American developers. They use the word "the community" when talking about the open source community, sprinkle their speech with *pro forma* negative comments about Windows, express dismay at my lack of familiarity with the "Fantastic Four" comic book series. For some, this cultural immersion comes as a result of participating in the world of software development. Others stress that they were "nerds" and preferred role-playing

games to soccer before they saw their first computer. In the majority of biographies it is impossible to separate the two processes: the deepening engagement with the US-lead nerd culture and with the profession of software develop usually go hand in hand. The software profession is seen as the ideal and often obvious career choice by many “nerds,” while fluency in the nerd culture is important for asserting membership in the profession.

Since the familiarity with this US-led culture is based on exposure to texts rather than oral conversations, the foreign culture is occasionally reproduced with an “accent.” For example, in one of the interviews my interviewee proudly referred to himself as a “geek” but pronounced the word with a soft “g” (“jeek”), a mistake hardly surprising for someone who had *seen* but not *heard* the word. This simple mistake in a foreign language would be hardly remarkable, were it not made as a part of self-description: “I am a jeek.” The English word “geek” is one of the few words the interviewee has to describe an affiliation that is quite obviously central to his view of himself.¹⁵ Such imperfect use illustrates the more general difficulties the developers experience in claiming membership, as such imperfect cultural performances suggests to the more central participants the liminal status of Brazilian developers. This questioning of status translates into “prejudice” that many of the Brazilian developers believe American companies show towards them (visible, for example, when it comes to getting contracts) and in turn reinforces the

15 Portuguesified version of English “nerd” (pronounced “nehji”) is used by many developers when speaking in Portuguese but is found offensive by others. The word “desenvolvedor de software” is accepted by most, but is seen as referring to the occupation rather than a culture. In other words, one *works* as a developer but one *is* a nerd. In this case, the interviewee is sufficiently proficient in English to know that “geek” is of often a preferred term to “nerd” in English, and therefore uses this word to describe himself.

After hearing the word pronounced this way another time, I mentioned this to another Brazilian developer. He expressed surprise at such pronunciation, saying that he thought everyone knew to say “geek” with a hard “g.” “How would one learn this?” I asked. “You hear it in the movies,” he responded. This second interaction highlights another important characteristic of the local situation: while some Brazilian developers pick up the right pronunciation from American movies and some do not, they are willing to assert association with a group label (“I am a geek”) that is rarely spoken in the local community.

importance of asserting membership.

To understand why Brazilian developers are so keen on asserting their links with the software world outside Brazil, we must appreciate the extent to which they draw on the power of the global software world. First, they depend on its practical power to solve specific problems faced by Brazilian organizations. Since the needs of Brazilian clients are often not unlike those in the United States, the developers find themselves in a position bring to Brazil foreign solutions to foreign problems which can be made to work in Brazil with some modifications. (Of course, part of the reason such imported solutions apply so well to local problems has to do with the fact that the local problems are themselves imported. The developer's task is often to provide a missing piece in a technological system already mostly built from foreign components and designed to support foreign practices.) Second, however, we must appreciate the extent to which Brazilian developers are selling to local organizations not just solutions to practical problems but also a "global" modern image, offering at a discount some of the prestige that could otherwise be obtained by having the organization's IT infrastructure modernized by IBM.¹⁶

Regardless of whether we stress the practical or the symbolic value of foreign information technology, Brazilian software developers find themselves playing a mediating role in its adoption, which requires that they successfully perform the role of *representatives* of the largely

¹⁶ The use of information technology as a modernizing strategy of course has long history. The first computer was brought to Brazil by IBM in late in 1950s in preparation for the 1960s census. Apart from the often discussed symbolic importance of the census as a symbol of a modern nation-state (Anderson 1991), this computerization must be considered in the context of a broader rush towards modernizing Brazil launched by President Juscelino Kubitschek in mid 1950s and symbolized best by the construction of new futuristic airplane-shaped capital Brasilia, inaugurated in 1960. The *symbolic* importance of the information technology has been pointed out in many accounts of computerization. For example, when describing the adoption of computers by the first Brazilian agencies, Dantas (1988) writes:

The computer was bought as an indispensable symbol of *status*. Excursions were promoted to show the powerful and mysterious machines, installed in glass boxes. The visitors were impressed by dozens of lights blinking and with the printers quickly spitting out mountains of paper containing information. It was quite an event. (p. 5 of chap. 4, my translation, original emphasis)

foreign software world. Acting as such representatives requires that their membership in this world is unproblematic. They must appear *native* in it. “You must not only walk the walk, but also talk the talk,” says another software professional. Anyone can learn to “walk the walk” (i.e., perform the actual tasks) after taking some time to learn, he explains, but not everyone can “talk the talk.” Yet, he says, “When the client or employer talks to you, they want to see that you are a guy who *lives* this, who is a part of this. And they just see it in your eyes.”

Contradictions

Striving towards full membership in the software world does not imply outright rejection of everything local. The developers realize the importance of the local community and even local knowledge. Both have to be handled carefully, however, in order to not undermine the links to the global world.

The local community is important for developers for at least several reasons. First, it is often a crucial source of jobs. Regardless of how successfully they adopt the culture and practice of American developers, American labor market remains closed to most of them. Furthermore, getting the local jobs and contracts often requires “*jeitinho*” - the exercise of personal and family connections and Brazilians consider a uniquely Brazilian tradition and which, in the view of many of them, flies in the face of the perfect meritocracy mandated by the global nerd culture.¹⁷

Second, interactions with local colleagues are an important source of information about foreign technology. The developers are careful to assert that they could *in principle* learn

¹⁷ While there is undoubtedly a difference in the amount of “*jeitinho*” between Brazil and the United States, it is my impression that many Brazilian developers overestimate the extent pure meritocracy north of the equator.

everything they want on the Internet, but can often save themselves time by using local sources. In other words, first hand knowledge is valuable, but it is also quite costly. A common compromise is to rely on others as an “indices” of foreign sources (Takhteyev 2005): when faced with a question one may ask for local advice, expecting, however, not a complete answer but “pointers” – often a keyword that can be entered into a search engine to retrieve the “original” sources. In fact, such “pointers” are sometimes *preferred* to more extended replies:

Celio: You go to the other person who knows the subject and you ask for pointers. “Where should I look?” I don’t want him or my professors to spend hours and hours talking about everything that I can *read*. I just don’t know *where* I can find it and they know. This is the point. “Where can I find it?” - “Here you can find it.” And then you go and read.

Handled this way, reliance on other people becomes a matter of pulling efforts for the purpose of keeping up with foreign sources (ideally with everyone doing their share) rather than just getting the information second hand.

While talking about resources such as online forums, the developers often highlight the informational properties of foreign forums, talking about them as “sources of knowledge”, while stressing the *social* side of local resources, describing them as communities but being careful not to over-stress the knowledge contained in such resources (Takhteyev 2007). Too much appreciation for foreign discussions as *communities* would question developers already strained commitment to the local community in Brazil. Stressing the “knowledge” contained in the local forums, on the other hand, may potentially suggest that the developers are getting their knowledge from those communities rather than directly from foreign sources and put in question their membership in the global software world.

Similarly careful footwork is required around the notion of local knowledge. The

developers occasionally realize that their knowledge of Brazilian context puts them in a better position to serve local needs and can be used as a weapon in competition against the multinationals. While asserting this advantage, they must be careful to not let this claim bring in question their position as software developers in a global sense. One of the interviewees (in this case a manager of a software company) does so by arguing for two “axes” of knowledge. One can separate the local contextual knowledge of business problems, he argues, from the universal technological knowledge. Pointing to the “two axes” of software knowledge in this way makes it possible to allow for local knowledge in areas it would be to the advantage of local developers (understanding the specific needs of local organizations), while stressing the universality of knowledge in areas where the local developers are potentially vulnerable (the technology itself).

Even with such solutions, however, many problems remain unsolved. While the local companies and developers are often forced to work together by virtue of being stuck in the same place, the symbolic power of global technology creates an incentive to actively avoid local resources when possible, undermining formation of the kind tight local networks that has helped the success of Silicon Valley (Saxenian 1996). In my second round of fieldwork I investigated the community formed around the Lua programming language. Developed since 1992 at PUC-Rio,¹⁸ Lua has been steadily growing in popularity in the United States, recently rising to the 15th position in the TIOBE ranking of programming languages.¹⁹ It’s use in Brazil appears to be minimal, however, especially outside the narrow network of graduates of PUC-Rio Informatics Department. While it’s use is starting to grow slowly now, those who use typically only look at Lua after reading positive reviews about it in foreign press. Before such reviews became wide

18 Pontifícia Universidade Católica do Rio de Janeiro

19 www.tiobe.com/tpci.htm

spread, however, the following reaction described by one developer (a former student of PUC-Rio) appeared to be typical:

Ricardo: I remember that we had to do a [class] project in C²⁰ and she [the professor] taught a new language, that just existed for a few years, invented at PUC and called “Lua.” I looked at that and was like: “Eew! A language invented here at PUC? What a silly idea! I am not going to learn this. I’ll never use it in my professional life! What will I do with it? And I remember there being two parts to the assignment that she sent us. One part was in C, another part in Lua. And a girl who was doing a part of the assignment with me... [I told her] “Here, do the part in Lua, because I am not going to learn this stuff, I don’t want to know about Lua. I’ll do the part in C, which is more interesting, since I’ll use it.”

In Ricardo’s case, through personal contacts he ended up joining a Lua startup (the only one to date in Rio to the best of my knowledge and the knowledge of Lua’s authors) and is working with Lua today. Those outside the immediate network (and most of those within in it), are only starting to consider Lua today, when it has proved itself abroad and when much of the early mover advantage has been lost to the Brazilian community.

Conclusion

Software developers working in Rio de Janeiro, Brazil, find themselves at a periphery of a global professional world in which they strive to assert membership. Claiming such membership and displaying their links to this global world is important both for competition against other local developers as well as for maintaining an image of proper representatives of the profession vis-a-vis both local clients and potential foreign collaborators. The display of developers’ global links, however, must be carefully balanced against the need to maintain commitment to the local community of developers, which can neither be fully embraced or rejected. Resulting tensions

²⁰ The most popular programming language in 1990s.

are partly alleviated through careful rhetorical work that places the foreign and the local into different domains making it easier for the developers to be simultaneously “local” and “cosmopolitan.” Yet problem remain. Foreign technology never fully seizes to be foreign and the local relationships often set a limit on the social engagement with foreign communities. At the same time, the symbolic power of foreign connections strains local relationships and even more so for the diffusion of local technology. Understanding the complexity of such dual allegiances becomes crucial for understanding the functioning of peripheral professional communities, not just in software, but also in other fields.

Acknowledgements

The author thanks the participants in the study for their time and patience and André Carregal for his thoughtful comments on this paper.

Bibliography

- Adler, E. (1987). *The Power of Ideology: The Quest for Technological Autonomy in Argentina and Brazil*. Berkeley: University of California Press.
- Anderson, B. (1991). *Imagined Communities: Reflections on the Origin and Spread of Nationalism*. Revised edition. London: Verso.
- Barley, S.R. & Kunda, G. (2004). *Gurus, Hired Guns, and Warm Bodies: Itinerant Experts in a Knowledge Economy*. Princeton, NJ: Princeton University Press.
- Brown, J.S. & P. Duguid (2000). *The Social Life of Information*. Boston, MA: Harvard Business School Press.
- Castells, M. (2000). *The Rise of the Network Society*. 2nd edition. Oxford: Blackwell.
- Dantas, V. (1988). *Guerrilha Tecnológica: A Verdadeira História da Política Nacional de Informática*, Rio de Janeiro: LTC. [Technological Guerrilla War: The True History of the National Informatics Policy, in Portuguese.]
- Duguid, P. (2005). “‘The Art of Knowing’: Social and Tacit Dimensions of Knowledge and the Limits of the Community of Practice,” *Information Society*, 21(2), pp. 109-118.
- Evans, P. (1979). *Dependent Development: The Alliance of Multinational, State, and Local*

- Capital in Brazil*. Princeton, NJ: Princeton University Press.
- Evans, P. (1995). *Embedded autonomy: states and industrial transformation*. Princeton, NJ: Princeton University Press.
- Knorr Cetina, K. (1999). *Epistemic Cultures: How the Sciences Make Knowledge*. Cambridge, MA: Harvard University Press.
- Kunda, G. (1992). *Engineering Culture: Control and Commitment in a High-Tech Corporation*. Temple University Press,
- Latour, B. (1988). *Science in Action: How to Follow Scientists and Engineers Through Society*. Cambridge, MA: Harvard University Press.
- Lave, J. and E. Wenger (1991). *Situated Learning: Legitimate Peripheral Participation*, Cambridge: Cambridge University Press.
- Levine, E. (1972). "Chicago's art world: the influence of status interests on its social and distribution systems." *Urban Life and Culture*, 1(3), pp. 293-322.
- Luzio, E. (1996). *The Microcomputer Industry in Brazil: The Case of a Protected High-Technology Industry*. Westport, Connecticut: Praeger.
- Marques, I. da C. (2003, May-August). "Minicomputadores brasileiros nos anos 1970: uma reserva de mercado democrática em meio ao autoritarismo." *História, Ciências, Saúde Manguinhos*, 10(2), pp. 657-81.
- Orr, J. (1996). *Talking About Machines: An Ethnography of a Modern Job*, Cornell University Press.
- Rogers, E. (1995). *Diffusion of Innovations*. 4th edition. New York: Free Press.
- Romer, P. (1993). "Idea gaps and object gaps in economic development." *Journal of Monetary Economics*, 32, pp. 543-573.
- 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.
- Shibutani, T. (1955). "Reference groups as perspectives." *American Journal of Sociology*, 60, pp. 562-569.
- Strauss, A. (1978). "A social world perspective." *Studies in Symbolic Interaction*, 1, pp. 119-128. Greenwich, CT: JAI Press.
- Strauss, A. (1984). "Social worlds and their segmentation processes." *Studies in Symbolic Interaction*, 5 (edited by Norman Denzin), pp. 123-39. Greenwich, CT: JAI Press.
- Takhteyev, Y. (2005). "Googling across the equator." http://takhteyev.org/papers/googling_across/
- Takhteyev, Y. (2007). "Using foreign forums." Proceedings of the Fortieth Hawai'i International Conference on System Sciences (HICSS-40). Los Alamitos: IEEE Press.
- Tigre, P.B. (1983). *Technology and Competition in the Brazilian Computer Industry*. London: Frances Pinter Publishers.
- Traweek, S. (1988/1992). *Beamtimes and Lifetimes: The World of High Energy Physicists*. Cambridge, MA: Harvard University Press.
- Unruh, D. (1980). "The nature of social worlds." *Pacific Sociological Review*, 23, pp. 271-296.
- Van Maanen, J. and S.R. Barley (1984). "Occupational communities: Culture and control in organizations." In B. M. Staw and L. L. Cummings (eds.), *Research in Organizational Behavior*, Greenwich, CT: JAI Press, v. 6, pp. 287-365.