Googling Across the Equator

Yuri Takhteyev

UC Berkeley School of Information

***Abstract*—The paper reports preliminary results of a 5 month interview study of learning practices among software developers in Rio de Janeiro, Brazil, focusing on the interaction between individual learning over the Internet and use of local social resources.**

> *Procure no Google (http://www.google.com) - É muito raro não encontrar algum esclarecimento de dúvidas no Google. Lembre-se: o Google é seu amigo. [...] Se você conseguir ler documentação escrita em inglês, vale a pena consultar também o histórico da lista Python-Users. O Google provê a indexação dessas páginas, portanto, basta usá-lo para fazer pesquisas. Vale a pena ainda verificar o site oficial do Python (http://www.python.org).*[1]
>
> The welcome message for the *python-brasil* mailing list, September 2005

I. Introduction

RECENT literature on economic development has increasingly stressed the emergence of "knowledge economy" - the state of the world where economic success is presumably determined not by possession of factors of material production or favorable location but by knowledge, skilled labor and beneficial positioning relative to the global "flows of information" (Castells, 1996). While quick absorption of new technical knowledge is hardly a panacea for socio-economic problems in developing countries, it is a necessary part of the solution. It is thus important to understand how new technical knowledge produced in the developed countries becomes available elsewhere in the world. This question has been typically approached looking at either institutional strategies for technology adoption or at diffusion of specific technologies. This paper presents preliminary results of a field study that looked at technology adoption by individual software developers in Rio de Janeiro, Brazil through a series of ethnographic interviews.

Learning how to apply new solutions to problems constitutes a key aspect of the work of modern software developers, in Rio de Janeiro just as well as in Silicon Valley. This learning is an active process and constitutes a substantial part of daily work of a software developer, and can be divided up into three types of activities: "scanning" (looking what's new), "searching" (looking for a solution to an existing problem), and "studying" (obtaining mastery of a specific technology). As this paper shows, all of those activities rely greatly on codified knowledge on the Internet. While social interactions play an important role in learning for many people, successful learning from colleagues often itself depends on the use of codified resources. One is expected to consult other people only after exhausting available textual resources and the result of asking a colleague is most often a *reference*, with an implicit or explicit advise to do an Internet search for details. It is thus important to not overstate the role of "tacit" knowledge and learning through participation (Lave and Wenger, 1994; Duguid, 2005) in such fields as software engineering, but instead to give full account of the role played by the Internet and foreign knowledge. At the same time, use of the Internet is often intertwined with use use of local resources and the paper aims to show how the two are combined.

The paper also discusses some of the preconditions for successful learning over the Internet, which include Internet connectivity, ability to read English, a certain amount of foundational knowledge, general learning skills and market demand for engineering knowledge.

II. Method

From July to December 2005 I conducted ethnographic interviews with 50 software professionals working in Rio de Janeiro to understand what knowledge they need to do their work and how they acquire that knowledge. Each interview lasted between 45 minutes and 2,5 hours and included discussion of education and career history, current job functions, and detailed examples of some of the things that the participants learned more recently, including cases of "important" and "trivial" learning. The interviewees were recruited using a theory-driven snowball sample, which aimed to include a wide range of work environments and levels of expertise. The sample thus included developers and system analysts working for small and large private software firms, public enterprises and universities. It also included a number of people writing software for academic research. I interviewed people with a wide range of expertise, from self-educated novices to computer science professors with doctoral degrees from outside Brazil. Most of the interviewees either had higher education or were nearing the end of a university program while working full-time. They attended a range of local universities, including PUC-Rio (which has Brazil's top ranked computer science program), local public universities, and lower-ranked for-profit private schools[2], though with a skew towards higher-ranked schools. Finally, while most of the interviewees were currently directly involved with writing software code, I interviewed a small

[1] "Look in Google (http://www.google.com) – It is very rare not to find some clarification for a question in Google. Remember: Google is your friend. [...] If you can read documentation written in English it is worth also consulting the archives of Python-Users list. Google provides an index of those pages, therefore it is sufficient to use Google for research. It is also worth checking the official Python site (http://www.python.org)." (Here and elsewhere the translation from Portuguese is mine, unless stated otherwise.)

[2] As a general rule, public universities in Brazil are seen as providing better education, with private schools taking those students who either could not pass exams into private university or needed to work full-time while studying. Several prestigious private catholic universities like PUC-Rio provide a notable exception from this rule.

number of people who have done development in the past but are currently working in managerial roles.

In addition to more formal interviews I've conducted less formal conversation with about 20 people involved with Rio software industry in other roles (e.g., funding agencies) to get a wider view of the industry.

The interviews were conducted in English or Portuguese depending on interviewees' level of comfort with spoken English[3].

III. Googling and Asking

Whether asked how they learned specific technologies, or about how they generally learn things nowadays, "the Internet" is by far the most common answer among the Rio developers that I interviewed. Apart from numerous casual references to Internet searches ("so I looked it up in Google"), many interviewees highlight the general importance of this resource. Summarizing our interview, one developer says:

> Alvaro: I think today that the Internet is a great vehicle and I use it to learn anything. If you mention something today, I don't know what, I go straight to my micro, Google – enter Google - and read. [...] If I want to learn more, beyond the Internet, I look for a book.[4]

Another developer, "Jaime" talks about the Internet as "the world's greatest library":

> Author: When you are working in your home and have some kind of question, about how to do something, what do you do?
>
> Jaime: Oh, today it's the Internet. Nowadays the world's biggest library is the Internet. The Internet today has all the imaginable and unimaginable resources.[5]

Like most engineer's over 30, Alvaro and Jaime not only talk about the importance of the Internet in their role, but point out the change it has brought, pointing to the difference between what they do "nowadays" and their earlier experience.

Like Alvaro, Jaime later talks about going to books for deeper knowledge. Many other participants prefer the Internet for nearly all situations or talk about books that can be downloaded off the Internet. It is important to note, however, the essential similarities shared by those two competing resources: (1) Both contain knowledge "codified" in textual form, (2) in both cases the original knowledge is typically produced and codified into text outside Brazil, and the text is usually written in English – a foreign language for all interviewed developers. The most important difference between them is the ease of access to the materials on the Internet when contrasted with books[6].

The participants report primarily using two types of resources on the Internet. One, type - "documentation" ("a documentação") - consists of materials maintained by people or organizations offering the technology. For example, Sun Microsystems maintains a website (http://java.sun.com/) which is used by many participants to learn more about Java. Smaller projects usually also offer documentation on their websites[7]. A particular type of documentation often mentioned by the participants is a "tutorial" - a text that is meant to guide learner's hands-on experimentation with the software.

The second type of resources is "the forums" ("os forums"). Unlike documentation, which is typically purposefully written to explain the new technology, forums represent a textual trace of an actual discussion about it. Unlike documentation, that must rely on developers' guesses as to what the learner needs (Orr, 1996), forums contain real questions from the users[8], usually answered either by the developers of the technology or by other users.

While quick to bring up examples of Internet use, few participants mention other people as a source of knowledge until asked about it explicitly. When asked, many do talk about the importance of "exchanging ideas" with others. Such exchange may occur formally or informally. "Claudio", who works with computer and software development training courses tells about a casual conversation with one of his friends:

> 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 [unclear] - those technologies that are coming up with cell phones. And we weren't trying to do anything with this – we were just playing billiards.[9]

For many interviewees (though far from all) such conversations are essential for staying oriented in the world of technology and knowing what is new, thus forming an important part of "scanning" aspect of learning. An experienced developer talks about simply not having time to follow all the written materials on his own:

> Author: Could you learn *just* with the Internet?

---

[3] As my Portuguese became more fluent over the 5 months of the study, my interviewees grew more reluctant to speak English to me. Thus, some of the interviewees that were conducted in English in August probably would have been conducted in Portuguese if they were to happen in November. All quotes used in this paper were verified by a native speaker of Portuguese.

[4] "Eu acho hoje que a Internet é um grande veículo e eu hoje uso ela, por examplo, para saber alguma coisa. Voce falar hoje alguma coisa, não sei o que é, ja vou para meu micro, google, tá, entra no Google, e leio. [...] Se quero me profundar alem do Internete, eu busco um livro."

[5] Author: Quando você está trabalhando em sua casa e você tem algum tipo de dúvida, sobre como fazer alguma coisa, o que você faz?

Jaime: Ah, hoje é a Internet. Hoje em dia a maior biblioteca mundial é a Internet. A internet hoje ela tem todos recusos imaginaveis e imaginaveis.

[6] One should note, however, that the Internet has also greatly facilitated access to printed books: most of my interviews order their books online from the United States and remark that apart from cost and delivery time, ordering finding books very easy.

[7] For an example of a smaller project mentioned by one of the interviewees, see Visual Python project page at http://vpython.org/.

[8] The users of such technology are of course themselves also developers of technology. Engineers build on each others' work and every developer is thus also a user of technology.

[9] "E a gente tava jogando, e o papo surgiu de tecnologia e a gente foi falando, de câmera digital. É... tanto de mega pixel... e ai tecnologia de compressão de arquivos... e a depois debandou pra... pra... celulares com câmera... e ai a gente foi falar de tecnologia celular. Ai falou de GSM, de CDMA, de CDMA [unclear], de é... E ai falou de [unclear]. Essas tecnologias que estão surgindo de celular. Ai não tinha nada a ver com isso... a gente tava jogando sinuca, só."

> Vinicius: I guess you could if you had the time to actually read... On the other hand, what do I do now - I read technical magazines, I buy some of them, and this is how you eventually get to know about new technologies that are becoming available. And again, talking with peers. There is no time to read everything. I don't have enough time to read all the magazines.

Note that rather than dismissing the importance of textual sources, Alberto stresses the importance of dividing up the work that is required to process the information in them. Many other developers, however, do attempt to do this on their own, sometimes then "broadcasting" their discoveries to others by email. In addition to magazines mentioned by Alberto (and many others), many developers mention using websites of large companies as a source of news (e.g., the same http://java.sun.com), or websites dedicated to technology news (e.g., http://slashdot.org).[10]

Friends and colleagues may similarly be helpful when looking for solutions *to specific problems*. When asked for an example of how he might ask other people for help, "Jaime" says:

> Jaime: There are two situations. Could be that I am sitting there programming and a problem shows up. "Hey, dude, already looked for it, couldn't find, can you help me?" He goes: "Of course." So he helps me. The other is sometimes 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." "A[illegible]ol. Great." So, I went back home – yay, it worked![11]

However, such specific questions are different from the casual chat described above by Claudio, and as Jaime's quote shows, one typically resorts to such requests only when when other methods fail. First, one typically searches on the Internet. Jaime makes sure to point out in his hypothetical conversation with a friend: "Already looked, didn't find."

The interviewees give at least three reasons why they might search on the Internet before talking to others: lack of local expertise, the social cost of asking others, and the ease of searching online.

1. Many of the experienced users often simply do not know anyone who knows more about the technology they are working with than they do themselves. After Alberto tells me about using a book to learn about Java cryptography, I ask him about other people working with this technology:

> Author: Did you know someone here with whom you could talk about this, about cryptography?
>
> Alberto: No, not this topic. Not this topic. There is nobody. Even "Rodrigo" [Alberto's knowledgeable close friend] knows a little, but not for example Java, for example - nobody has experience with Java.[12]

Alberto says he does end up talking to "Rodrigo" about Java cryptography because "Rodrigo is smart" and Alberto finds that "always helps" to talk to him about things he has read. However, this is hardly a case of an apprentice learning from an expert: Alberto is certain that he knows more about the topic than his friend.

2. In cases where experts *are* available locally (more common for novices), there is usually substantial social cost associated with interrupting their work or introducing a work problem at an otherwise social event. When asked to explain why he said he doesn't ask his colleagues about "simple things," a developer says:

> Cicero: If I already asked him two questions in one day, both super-simple stuff, then I probably wouldn't ask the third, not to interrupt the guy every hour.[13]

Many other participants similarly report thinking how many times they can interrupt others with questions. Prior research on the Internet also allows them to get more value out of a single question:

> Luciano: [illegible] if I had somebody to talk, I would start with Google, then ask for help.
>
> Author: Why?
>
> Luciano: I think to get more prepared for receiving the information through other people.
>
> Author: What do you mean by "more prepared"?
>
> Luciano: To know what he is talking about. Some concepts he might talk... to get acquainted with the concepts. For example, "frequency," "sampling", for example. And when he tries to explain something to me he might use concepts that I must be acquainted to, the technical terms he is talking about.

Another interviewee says:

> Alexandre: I guess I am the kind of person who does not want to bother anyone, unless... unless I have tried to do it myself. I think it's more profitable that way.
>
> Author: What do you mean by "more profitable"?
>
> Alexandre: I can ask the right questions and I can get more of their answers.
>
> Author: If you try looking for it before hand? Can you explain it?

[10] Those who read Slashdot report spending up to 2 hours a day reading news. There thus seems to be a general consensus that one *can* keep track of technology using the web, though doing so can be time-consuming.

[11] Jaime: Eh, são duas situações. Pode ser que, por exemplo, eu estou lá programando, aí pintou uma dificuldade. "Fulano, pô, tem né, ja pesquisei, não sei o que, não consegui achar, pode me ajudar? Ele: "Pois não." Aí ele me ajuda. A outra seria as vezes a gente senta pra almoçar. Aí é aquele bate papo informal, não sei o que, um conversa, um fala familia, outro fala da mulher, e surge o papo de "Pô, você..., pô outro dia quebrei a cabeça fazendo uma rotina tal. Pô, 'cê tem idéia de como é que faz?" Ai o cara "Ih não, ja fiz essa rotina, usa biblioteca tal." "Ah, legal, beleza." Chegei em casa – ih, funcionou!

[12] Author: Você sabia algumas pessoas aqui com quem você poderia falar sobre isso, sobre criptografia?

Alberto: Não, esse asunto não. Esse asunto não. Não tem ningem... Até o "Rodrigo" sabe um pouquinho, mas não, por exemplo, Java, por exemplo – ele não tem pratica em Java.

[13] Cicero: Se eu fiz ja duas perguntas pra ele no dia, coisas hiper simples talvez não faça a terceira, pra ficar parando a cara toda hora tambem.

> Alexandre: Well, I'll say "I am having a hard time trying to do something" and they would say "Well, have you tried *that?*" and I would say "Well, I tried and that happened or it didn't work because of something." Otherwise they would say "Have you tried that?" and I would say "No," and have to go back and then try it, go back to them and say "Now I tried, it didn't work."

This reluctance to "bother" others with questions reflects the common perception that search for solutions is an integral part of software development, rather than an occasional side task. Most interviewees have regular meetings with colleagues to achieve a general agreement as to what approach to use and to discuss some particularly difficult problems. By the end of the meeting, however, the remaining problems are usually divided up among the developers. Once the tasks are subdivided, the engineers work *individually* to resolve their share of the problems. There is thus a thin line between making use of colleagues' expertise to do your work and asking them to do your work for you. Most of the interviewees walk this line carefully. They do consult their colleagues when they get "stuck", but make sure they do their share of work first.

3. There are notable exceptions. "Jorge," a computer science professor, talks about consulting students without hesitation. He says he has each student investigate a specific research direction, and when he has any technical questions he often knows students who he could ask about details. However, Jorge has a lucky combination that is quite rare: a pool of experts, whom he carefully reared, vis-à-vis whom he has unlimited social capital, and who are thus at his disposal. Such situation is uncommon since novices usually face high social cost when asking for advice while experts often have no one to assist them locally.

Even Jorge, however, uses the Internet daily in his research. This may happen in cases when he is investigating a new area in which none of his students have gained an expertise yet. Even more often, however, it is simply a matter of *convenience*. Another interviewee, when asked whether he usually asks friends or searches on the Internet, replies with a rhetorical question: "What is easier? Search on the Internet." For many kinds of questions, the interviewees say that while asking a co-worker who shares a desk may be easier than doing an Internet search, getting up and walking to another room or picking up a phone already requires too much effort[14].

IV. Trading References, Looking up Details

Interviewed software developers routinely rely on the Internet to learn new ways of solving the problems they face at work. They also routinely share this new knowledge with local colleagues, either when asked for advice or actively "broadcasting" interesting discoveries by email. Such exchange is important, even if it is overshadowed by individual research online. Such sharing, however, typically relies on exchange of "references" with an expectation that the recipient will get the details through a textual document on the Internet.

> Vinicius: Eventually they would say, hey [Vinicius], have you heard the new, I don't know, language that is coming out. Or this new technology that Microsoft, or Intel, or Sun is releasing. And so if it is something that clicks, if it is something I have interest in, you go and try to find information.

Thus, the way to learn more about something mentioned by a colleague is not to talk to them longer, but rather to "go and try to find information," which means searching on the Internet. Vinicius, now in late 40s, contrasts his current experience with mid 1980s:

> 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 get *any* kind of information you want on the Internet.

A reference may include a URL of a specific document on the Web. In oral communication, however, it is usually enough to provide a "keyword" - a phrase that one can use for an Internet query. Keywords may be provided with or without specific documents in mind and in many cases, it is in fact ambiguous.

> Wanderley: And then another colleague, one of those colleagues said: "Well, you can use JMeter, it's a project inside Apache Group, Jakarta Apache." And then, well, I have already heard about JMeter. And then I put "JMeter" into Google, of course. One of the first items that occurred in Google was Jakarta Apache JMeter page and then I started to look for more information about JMeter.
>
> Author: On that website?
>
> Wanderley: Using Google, using Google.

It is hard to say in this case whether the colleague who suggested "JMeter" to Wanderley intended him to specifically go to the JMeter project website, which is the first result when searching in Google for "jmeter". Rather, there is a shared understanding that knowing the term "JMeter" Wanderley is fully capable of finding the details on the Web. Putting JMeter to use may not be altogether trivial, as Wanderley later explains. Those difficulties, however, can be resolved individual search.

Thus, a common way of making the most out of other people's time is to "sandwich" a personal interaction between uses of the Internet: one first goes on the Web to acquire enough background knowledge, then asks the expert, then returns to the Internet armed with new keywords.

V. Preconditions for Learning

Most participants thus agree that textual materials provide the key to the knowledge they need. Learning through such text, however, requires certain resources. In case of books, the main problem is money: while "importing" books through Amazon is technically easy, the cost is substantial, which leads many interviewees to either carefully plan what books

[14] Some interviewees report using instant messaging to talk to friends and colleagues who are not immediately next to them. Even so, searching on the Internet is often seen as an easier solution.

to buy, or to avoid books for the sake of free materials on the Internet. The use of the latter requires only an Internet connection, which, is easily available to most of the participants, who typically have broadband access both at work and at home. With the exception of one participant, who reported using dial-up from home, network connectivity is a non-issue for the interviewees and I will thus not discuss it here. Beyond physical access to resources, however, one need the ability to actually learn from knowledge represented in a technical text.

A.English

Both electronic texts and books, are primarily available in English. Software developers I interviewed varied substantially in their level of oral English proficiency: Some spoke flawless English, others said they could speak English but preferred to conduct an interview in Portuguese, yet others told me flatly that they could speak only Portuguese. At the same time, all reported successfully reading technical English, which they claim was necessary in their work, since on the Internet "it's all in English." In fact, ability to read English is expected by employers: a manager I interviewed lamented about mistakenly hiring a developer who could not read English. "That's terrible for someone who is supposed to work in computer science," he said,[15] adding that he had since introduced an English test for all new job candidates.

While all interviewed engineers agree that most information they need is available only in English, they vary in their attitude towards materials in Portuguese. Some express clear preference for materials in their native language and try to find them before looking for English. Others say the come across Portuguese resources occasionally and look at them out of curiosity, but never search for them. Others avoid them. One developer explains (in Portuguese) why he does not use Portuguese mailing list:

> Mario: Because the software community in Brazil is a hundred times smaller than abroad and those lists [abroad], they include the very developers of the products, who often answer many questions. 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.[16]

Many other interviewees explicitly deny having such "prejudice" against resources in Portuguese but say that English materials are simply easier to find.

The need for English depends on the novelty of technology, as one of the participants explains (and a visit to any bookstore in Rio confirms), it is easier to find Portuguese texts for older, better diffused technology:

> Tiago: Depends. You have some people... when people know about technology just for the job, generally those are people who start with Microsoft, Visual Basic, they don't care too much about English because everything is in Portuguese for them, English is just for codes. And even then the labels are in Portuguese, all in Portuguese. The tutorials are all in Portuguese. In Python you can't rely on that. Today is better, but you can't rely on that. Java you can't rely on that too.

This comment perhaps explains why most developers feel that they can read *enough* English despite the fact that their level of English proficiency varies substantially (even in reading). Increased language capacity allows them to engage with newer technology (like Java or Python), which in turn increases demand on their English.

B.Basic Education

Most of the interviewees have a college degree in software engineering or a related field and started working during their second or third year in college. Of those who do not have a degree, all but one are currently enrolled in a program, typically attending classes at night. (A smaller but significant number have Masters degree or are currently working on one.) Some report going to college just to get "the piece of paper" expected by employers. Most, however, attribute at least some value to their university education, even if they claim to be largely self-educated ("autodidato") and are critical of their university education[17]. Three things come up most often: the importance of access to professors at the early stage of learning, the value of a curriculum, and learning how to learn.

Many interviewees say the university gave them "the basics" of computer science. When asked why they couldn't have learned those "basics" on their own (as they say they learned most things), two replies are common. Some feel that one needs a certain foundation of knowledge before one can learn alone and that at this phase an opportunity to ask questions is important:

> Celio: No, you can find the material, this is true. But even people who have very big IQ, it is very difficult to learn this stuff by yourself without discussing it. Sometimes you learn from the teacher – not from what he is writing on the board but from the questions that you ask and the answers that you get. This is the best way to learn.

Others, however, feel that they *could* have learned the material on their own, or even assert that this is what they *did*. Without a formal program, however, they might not have known what it is that they need to know. When asked why he couldn't have learned alone what he learned in his Masters program, one of the participants says:

> Alberto: Because you end up learning only what you like to learn, what you want. Or what you need to learn for

[15] This quote, originally in English and a part of a mixed English-Portuguese interview, has been edited for grammar. While the speaker stresses importance of reading and writing English, his own ability to *speak* English is limited, highlighting once again the difference in need for oral and written skills.

[16] Mario: Porque a comunidade de software no Brasil e centenas de vezes menor do que na fora e normalmente estas listas de discusão, elas involvem os proprios desinvolvedores do produtos e eles custumam responder bastante coisas. E as listas em português normalmente são destinadas... Pode ser um preconceito meu, mas normalmente contem mais pessoas que estão nestas listas porque não sabem falar inglês. Porque é o único recurso que elas têm.

[17] My sample included graduates of PUC-Rio (a private Catholic university), UFRJ and UFF (two federal universities), UERJ (a state university) and several smaller private universities (which typically rank lower than the four schools mentioned above). Only PUC-Rio graduates were not critical of their programs. Students who went to public schools complain about frequent absenteeism among the professors, while graduates of lower-ranked private schools cite lack of expertise among the faculty.

> day-to-day needs. When I did Master's I had to learn what I liked and what I didn't like. And discovered that that there were other things that I later came to like. That is, you are helped to learn. You are learning new things, you are stimulated in another way.[18]

The value of the university for Alberto is thus the *curriculum*, which "stimulates" one to learn. Another interviewee, speaks about the value of the curriculum explicitly:

> Celio: When you go to the university you have a defined curriculum that you have to go through, whether you like it or not. When you are on your own you don't know all the many subjects that are available, so you don't know *what* to look for. In the university they bring you some base, and then you can go through many other steps. Well, now I know that we have X, Y and Z. I really like X. Let's see what else I can find out about X.

To illustrate this point, Celio tells me about deciding to learn more about Java in 1997, at the time when few people at his university even new it existed. While prompted to study by the university (he needed to write a report about a programming language), Celio could not rely on local resources for actual learning. He thus learned Java using Internet and books and later taught a weekend course for professors at his university.

Just as often, however, the interviewees stress that above all, the university taught them how to learn. Angela, a graduate of PUC-Rio (Rio's most prestigious computer science program) says:

> Angela: The thing is, I spent 5 years in college. I learned three things that I think were worth learning. The first thing was to *think* and to relate things and analyze them like when you are doing calculus and physics. I don't use it, but I learned how to think and how to relate things. How to think - and this I think is worth learning. The second thing I learned is how to learn, and how to study and learn by myself. And the third thing I learned was not to be afraid of anything. I mean, someone might say "You must do it," I say "OK". I start thinking about the problem, instead of saying "Oh my God!" That's what I learned in college. Because everything I learned there, it's not valid anymore. I graduated in 1994, which is eleven years ago, and nobody uses the tools I learned then.

Like many others Angela points out the importance of general learning skills she acquired in college. Given the pace of technological change experienced by the interviewees and the fact that codified knowledge on the Internet potentially allows them to keep up with the changing technology, it is hardly surprising that knowing how to quickly absorb the knowledge contained in such texts becomes one of the most valuable skills that a university can offer[19].

[18] Alberto: Porque você acaba... aprende só o que você gosta, que você quer. Ou o que você precisa aprender na urgencia dia-a-dia. Quando eu fiz o mestrado eu tive que aprender o que gostava e que não gostava. E descobri tambem que existiam outras coisas que depois eu vim a gostar. Quer dizer, você é ajudado a estudar. Você esta aprendendo coisas novas, você é estimulado de uma outra forma.

[19] Ironically, such skills are not explicitly *taught* in most universities, the students discover them while learning other things.

VI. Demand

I've argued so far that most of the knowledge used by software engineers in Brazil is directly "imported" from abroad (individually by each person) through documents on the Internet or through books (less and less often). Knowledge created within the company is important for those working for large multi-national corporations[20], but plays limited role for those working for Brazilian companies. At the same time, the local software firms play a crucial role in learning, as they *pay* developers to solve problems. It is while solving those local problems that they absorb new foreign knowledge:

> Author: And this particular technology, have you since learned more about it?
>
> Wanderley: Not very much, almost nothing. Because... This is my particular behavior: if there is no practical thing to do, there is no problem to solve, normally I don't go and try to learn something without having a concrete use, something that I must solve.

Wanderley's focus on problems at hand is common. Other engineers try to "look ahead" to discover and learn today the technology that they might need tomorrow. When possible, they try to find someone who would pay for their learning effort:

> Jaime: So, a company calls you to work with a data base. Ah, I don't know how to do it. And now what, you are going to miss out on it? Or maybe you will try to assume the risk to learn. Because the game goes like this: I think when you accept a project, you aren't just accepting that project, right? That project can open doors to many more, understand. Other new things. So, you can never say no to a particular project. So there is a great need to stay on top of things.[21]

Many other interviewees similarly tell with enthusiasm about projects that allowed them to work with new and unfamiliar technology, as such projects gave them an opportunity to learn new skills. Lacking such opportunities they may learn new technology at home on their own time. Even so, most keep their eyes on the job market, and try to invest their time in technologies that they think will provide employment in future, much like Becker (1962) would expect.

Modern computer technology is sufficiently complicated, explains one of my interviewees, that while one theoretically can learn it alone, the investment into knowledge would be too costly to do it as a hobby. You thus need to do it

[20] Engineers working at some of the larger multi-nationals report that their corporate *intranet* sometimes contains better resources than the public Internet.

[21] Jaime: E ai a empresa chama para você para trabalhar com banco de dados. Ah, não sei trabalhar! E ai, você vai, entendeu, não vai deixar de pegar? Ou de repente você pode tentar assumir o risco de aprender. Porque o jogada é o seguinte: eu acho que quando você aceita um projeto, na realidade você não está aceitando somente aquele projeto, né? Aquele projeto pode abrir as portas pra outros mais, entendeu? Outras coisas novas. Então, nunca é bom você dizer não, pra determinado projeto. Daí a grande necessidade de você estar sempre sabendo.

professionally, for pay. Availability of qualified labor thus appears to be an *effect*, rather than a precondition of a software industry in a country like Brazil. (Though, good general education, English skills and network access are necessary if the labor market is to respond to such a demand.)

This insight helps us understand the lasting effect of Brazil's Market Reserve policy in 1980s, which created a computer industry in the country (Evans, 1995; da Costa Marques, 2003). While little of the technical knowledge from that era is used today, Brazil now benefits from the continuing industry, a culture of software use (especially by the federal government), and the idea that software can be developed locally. Brazilian software industry today still follows the business models it learned during the Market Reserve. The overwhelming majority of Brazilian developers (and of my interviewees) are employed by companies that provide software services for Brazilian clients, typically local (Veloso et al, 2003).

VII. Other Domains of Knowledge

This paper has so far focused on technical knowledge, which, I've argued, individual workers increasingly "import" directly via the Internet. It is not my intention, however, to argue that *all* knowledge related to software engineering can be acquired this way or that technical knowledge by itself is sufficient for developing a thriving software industry. In fact, successful software industry requires knowledge in at least two other areas, and such knowledge appears to be much harder to learn on the Web.

One domain is the organizational side of software engineering, the art of *managing* software development. Software development projects are typically too large for a single person, so they must be subdivided between several developers, whose work then needs to be coordinated. Organizing a large project thus requires complex organizational skills, both on behalf of managers and employees (Brooks, 1978). While few industry insiders in Brazil find Brazilian engineers lacking when it comes to technical expertise, many complain about lack of *discipline* – the ability to deliver work on time and to specifications. While many attribute such lack of discipline to "Brazilian culture," it may also be a sign of limited organizational skills among the managers. When good developers are promoted to management positions (a common practice in Brazil), they face questions that cannot be easily answered on the Internet, either because the resources are much more limited, or because they require substantially higher English proficiency.

> Afonso: To do the work I do now it is more important to know how to manage people. This is very difficult. The technical part isn't so complicated. But to lead people, manage people is the complicated thing. Very complicated. Not only to lead the people who report to you but also people who you report to. This is very complicated.
>
> Author: And how did you learn that?
>
> Afonso: Well, I read some books in psychology and in... I don't know how to say... sociology. But the main thing is my own day-to-day experience.[22]

It is likely that useful resources are hard to find because management knowledge is harder to codify than the more technical knowledge.

Software industry similarly requires higher-level business knowledge, including the knowledge of how to *sell* software and software services. Since few of my interviewees were involved with business aspects of software, I have no data on learning in this domain. However, talking to industry insiders leaves a clear impression that while Brazilian software engineers inhabit the same *technical* universe as their peers in California, Brazilian software businessmen are a world apart from their counterparts in Silicon Valley[23].

VIII. A Disembodied Practice

My interviewees' stress on of the role of the Internet for everyday learning contrasts with the emphasis on "tacit" and "local" knowledge in much of contemporary discussion of of the knowledge economy. The participants do mention the importance of personal experience with the technology. Some things, they say, cannot be be learned through reading: you have to actually *do* it. They often say they learn by "reading and tinkering" ("lendo e mexendo"). They stress, however, *personal, first-hand* experience acquired through *individual* work, not collective tacit knowledge that one absorbs by "being there" and observing work or others.

The lack of importance attributed to learning through embodied participation may be a reflection of the *kind* of knowledge work that software developers do. Unlike Latour and Woolgar's (1979**)** biologists who share their space with numerous physical "inscription devices," or Collins' (2001) physicist who must see their Soviet colleague's way of waxing a thread, software developers - in Brazil just like in California – work little with physical objects apart from their computers. Their work is a disembodied, textual art. Without closely looking at their screen and keyboard, one merely sees them typing on their keyboards and looking at their monitors and can learn little about how they do their work. Such close observation is rare, however – software coding is typically done individually.

In most organizations, the software code repository (typically hosted on a central server and accessible to all) works as a virtual environment in which the objects of work reside and can be observed. However, most often, engineers work on their separate parts of the project and rarely examine in detail

---

[22] Afonso: Para fazer o trabalho que faco agora é mais importante saber como gerenciar pessoas. Isso é coisa muito dificil. A parte tecnica não é tão complicada assim. Mas lidar com pessoas, gerenciar pessoas é uma coisa complicada. Muito complicada. Não apenas lidar com as pessoas que estão subordinados a você mas tambem lidar com as pessoas a quem você esta subordinado. Isso é bem complicado.
Author: E como você aprendeu isso?
Afonso: Olha, (unclear), ja li algums livros de psicologia e algums livros de... não sei como clasificaria isso, sociologia talvez. Mas a coisa mesma e a dia-a-dia experiencia propria.

[23] This perhaps is not surprising given the differences in business culture *within United States* reported by Saxenian (1996).

code written by their colleagues[24]. Thus, while some interviewees report learning a lot from colleague's code, they seem to represent a clear minority. Of the remaining interviewees, many do report learning from code, but talk about code written outside the company (and usually outside Brazil), available on the Internet. This highlights an important distinction between the physical environment that hosts the artifacts of Collins' physicists and the software repositories: while physical research labs typically have rather tight access policies, there are many repositories of source code that have their doors wide open. This may reflect both a cultural difference (the idea of "free software" and "open source" that has captured the minds of many engineers), as well as a practical difference: software repositories can be "visited" at no cost and such visits do not disturb the work that occurs in them.

IX. Conclusion

Codified knowledge represented on foreign websites plays a crucial role in technical learning for software developers in Rio de Janeiro. Internet does not merely *promise* to transform their work – it has already done so. Direct "import" of foreign knowledge in textual form requires that we rethink the role of "local knowledge" and *local* communities of practice, at least for some types of knowledge work. The fact that this importation of knowledge is done directly by *individual workers* means that we need to also rethink the role of the firm in the international transfer of technical knowledge. Rather then being a place where workers learn through participation in a living, embodied practice, the firm becomes a *stimulant* for learning by providing problems and offering money for solving them. It also serves as a social space where engineers can trade "references" to resources and concepts found on the Internet (even as they return to the Internet to learn the details). Abundance of technical knowledge in text similarly changes the role of the university, whose main role becomes to stimulate learning and provide an environment where remote texts can be discussed.

References

[1] Becker, G. S. (1962) "Investment in human capital: a theoretical analysis," *The Journal of Political Economy*, 70(5).

[2] Brooks, F.P. (1978) *The Mythical Man Month: Essays on Software Engineering*, Reading, MA: Addison-Wesley.

[3] Castells, M. (1996) *The Rise of the Network Society*, NY: Blackwell.

[4] Collins, H. M. (2001) "Tacit Knowledge, Trust and the Q of Sapphire", *Social Studies of Science*, 31(1), pp. 71-85.

[5] da Costa Marques, I. (2003) "Minicomputadores brasileiros nos anos 1970: uma reserva de mercado democrática em meio ao autoritarismo." *História, Ciências, Saúde Manguinhos* 10(2), p. 657-81.

[6] Duguid, P. (2005) "'The Art of Knowing': Social and tacit dimensions of knowledge and the limits of the community of practice," *The Information Society*, 21(2), pp. 109-118.

[7] Evans, P. (1995) *Embedded Autonomy: States and Industrial Transformation.* Princeton University Press: Princeton, New Jersey.

[8] Latour, B. and S. Woolgar (1979). *Laboratory Life : The Construction of Scientific Facts*. Thousand Oaks, CA: SAGE.

[9] Lave, J., E. Wenger (1991) *Situated Learning: Legitimate Peripheral Participation*, Cambridge: Cambridge University Press.

[10] Orr, J. (1996) *Talking About Machines: An Ethnography of a Modern Job*, Cornell University Press.

[11] Saxenian, A. (1996) *Regional Advantage: Culture and Competition in Silicon Valley and Route 128*, Harvard University Press.

[12] Veloso, F., A.J. Botelho, T. Tschang and A. Amsden (2003) "Slicing the knowledge-based economy in Brazil, China and India: a tale of three software industries,"SOFTEX/MIT, September 2003.

[13] Wells, D. (2004) "Collective Code Ownership" Available at http://www.extremeprogramming.org/rules/collective.html. Last retrieved Nov. 30, 2005.

[24] A new movement in software development, known as "Extreme Programming" or "XP," suggests that all engineers should work on *all* parts of the system, specifically for the sake of diffusing knowledge about it. This aspect of Extreme Programming methodology is referred to as "Collective Code Ownership" (Wells, 2004). Extreme Programming is yet to gain dominance, either in Brazil or in California.