Software Gigging: A Grounded Theory of Online Software Development Freelancing

Completed Research Paper

Raymund Sison
De La Salle University
Manila, Philippines
raymund.sison@delasalle.ph

Rabby Lavilles
MSU-Iligan Institute of Technology
Iligan City, Philippines
rabby_lavilles@dlsu.edu.ph

Abstract

Despite the rapid growth of online freelancing as an industrial sector, there is a lack of research in it, especially in a country like the Philippines, which, though an important online freelancing destination, belongs to the global south. This study uses the grounded theory method to explore the practices of online software development freelancers, or software e-lancers, in the said country. The result is a substantive theory that views earning flexibly *as the main concern of software e-lancers. To resolve this main concern, they engage in what we call* software gigging*, which is a cyclical process of* gig-hunting *and* gig-executing*, each of which has several activities. Software gigging becomes increasingly lucrative and flexible as one advances through the gigging stages of* noob, rockstar, and super-rockstar. *As they do software gigging, software e-lancers use various strategies such as* reputationing *or* circumtechventing*, depending on the specific gigging activity and stage that they are in.*

Keywords: Online freelancing, software development, grounded theory, gigging

Introduction

Online outsourcing, which involves the often offshored contracting of business tasks to third-party workers via the Internet, is growing rapidly as an industry, both in the number of projects and tasks posted on online outsourcing platforms (Kässi and Lehdonvirta 2016) and in the weighted average of the compound annual growth rates of these online labor marketplaces (Kuek et al. 2015). *Crowdsourcing* and *online freelancing* are two sectors of this industry; but though the former appears to have become a mainstream topic, research in the latter remains nascent (Lacity et al. 2017). A recent scoping review of peer-reviewed gig economy articles from January 2000 to July 2017 from six academic databases, including Scopus and WoS, similarly notes that "the biggest (research) gaps are a lack of studies looking at the lived experiences of gig workers" (Bajwa et al. 2018 p. 21).

According to a recent World Bank-commissioned report (Kuek et al. 2015), the top three online outsourcing destinations worldwide are the United States, India, and the Philippines, with 23.9%, 21.5%, and 18.6% of global workers coming from these three countries. However, when population size and the size of the labor force are taken into account, the Philippines rises to the top, with 2% and 4.8% of its population and total labor force, respectively, engaged in online outsourcing. Unfortunately, there is a dearth of research in online outsourcing in this country. A recent updated review of IT and business process outsourcing (Lacity et al. 2017) calls for more studies of other outsourcing destinations besides India, citing as one reason the caveat that practices that are effective in one context (e.g., Western client-Indian provider) might not be as effective in other contexts.

In order to help address the aforementioned need for more research in online freelancing, especially in a country that has become an important online freelancing destination, this paper looks into the concerns and practices of online freelancers in the Philippines. It focuses on online software development freelancing, which we call *software e-lancing* for short, which includes web development, mobile development, and game development. Among all categories, web and mobile development had the most earnings in Asia in the Elance-oDesk (now Upwork) freelancing platform in 2014 (Cunningham 2015).

The grounded theory method (GTM) is used in this study to understand the main concern of software e-lancers, and how they resolve this concern. GTM is recommended when the investigation involves nascent phenomena (Van de Ven as cited in Urquhart and Fernandez 2013). Its use in information systems (IS) theory development spans almost three decades, with (Orlikowski 1993) being among the seminal works. Special issues in major information systems (IS) journals have been devoted to the use of GTM in IS, and papers have been written that provide empirically based guidelines for using GTM in IS research (e.g., Urquhart, Lehmann, and Myers 2010).

The Glaserian approach to GTM (see e.g., Glaser 1998), which this study uses, eschews a "literature review in the substantive area and related areas where the research is to be done" (Glaser 1998 p. 67). This study, therefore, did not begin with any literature review, and consequently, did not have any research questions or theoretical perspectives at the start. Instead, the question that we tried to get answers to was a "grand tour" question (Simmons 2011), "designed to convey to the respondent that they are being invited to discuss what is relevant to them about the general topic area, on their terms" (p. 23). Our grand tour question was, "What is your main concern as an online software development freelancer, and how do you resolve it?" A substantive theory that answers these two questions will be presented and discussed in the succeeding chapters.

The rest of this paper is organized as follows. First, we describe the GTM used in the study. Next, we describe and discuss a substantive theory of online software development freelancing, which we call *software gigging*. Finally we summarize and provide some implications for practice and for formal theory.

Method

We begin this section by quoting one of the originators of the grounded theory method (GTM), Barney Glaser (1998, p. 115):

> "Grounded theory accounts for the action in a *substantive area*. In order to accomplish this goal, grounded theory tries to understand the action in a substantive area from the point of view of the *actors* involved. This understanding revolves around the *main concern* of the participants whose behavior continually resolves their concern. Their continual resolving is the *core variable*…It emerges as the overriding pattern. Thus the goal for a research using grounded theory is to discover the core variable as it resolves the main concern."

The above sentences were quoted verbatim because they contain important GTM keywords that we will be using in the methodological description that follows.

The Substantive Area

The substantive area in this study is online software development freelancing, which we call software e-lancing for short. The actors in this substantive area do software development work online, and they do so as independent contractors, not as employees of a software development company. The study focuses on software e-lancers based in the Philippines and whose clients are abroad. 52 actors meeting these criteria were interviewed. 71% of the study participants turned out to be single; 33% were female. 83% considered themselves full-time online workers; and 60% had bachelor's degrees in computer science or related fields. Their average age was 27 years old, and their average number of years of work experience was 5.

Phase 1: Theory Origination

GTM can be viewed as having three phases—theory origination, theory saturation, and theory elucidation—each of which can be described in terms of input, process, and output (Figure 1, adapted from Sison 2017). Glaserian GTM begins with the researcher's laying aside any theories about the substantive area before entering the field. This does not mean forgetting all that one knows about the area, as if that were possible; rather, it means "setting aside known theories for potential future comparison, which are done only if the analysis of the data indicates the relevance of these theories" (Urquhart and Fernandez 2013).

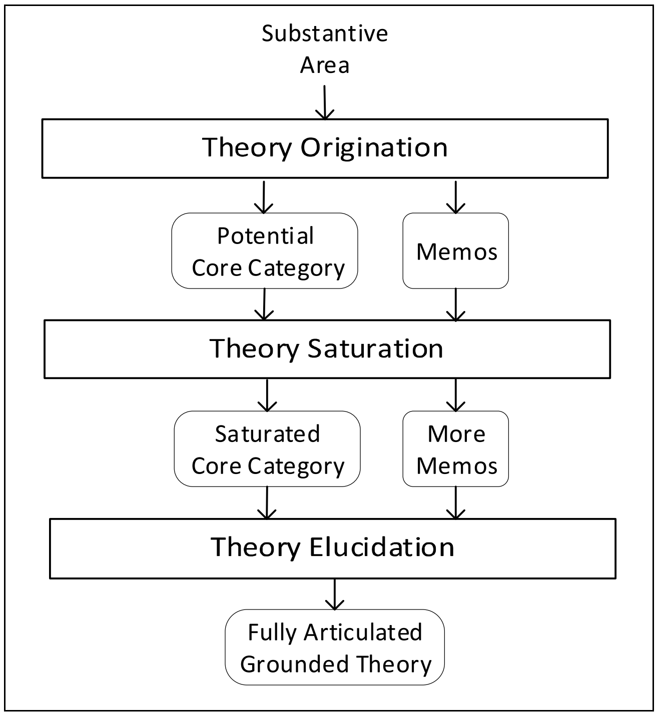

Figure 1. Three Stages of GTM

Theory origination is a cyclical process of data collection, coding via constant comparison, and memoing. Data collection in GTM is usually done through interviews. In this study, the interviews, which lasted 1 hour and 29 minutes on the average, were conducted face to face from October 2015 to December 2017. These were followed by clarificatory calls and instant message exchanges that lasted until March 2018. The interviews could be grouped into phases, as shown in the Table 1. The basic questions that were asked in the interviews were: How did you start working online? How would you describe your daily routine? What difficulties have you encountered and how did you deal with them? What do you like or dislike about your online job? In one interview, there would be so much more follow-up questions, which depended on the answers of the participants to the basic questions, and on our growing understanding of their main concern and how they resolve this.

Participants	**Dates**	**Interview Purpose**
1-8	October-December 2015	Pre-GTM exploration of online workers' experiences
9-32	January-December 2016	Theory origination (*transitioning* and *reputationing* as potential core categories)
33-41	January-April 2017	Theory origination (*gigging* as core category)
42-52	May-December 2017	Theory saturation

Table 1. Interview Phases

GTM data is, however, not limited to field notes or transcripts of interviews; in GTM, all is data (Glaser 2002, p. 145). As shown in Table 2, additional sources of data in this study included participant observation and participants' posts in relevant social media (e.g,. GitHub, LinkedIn) and online marketplaces (e.g., Upwork, onlinejobs.ph). All data were obtained and used only upon the informed consent of each participant.

Non-Interview Data	Source(s)	Purpose
Information about participants	OLMs LinkedIn Facebook Profile Referrals	Search for potential participants
Social media posts	Facebook, Instagram	Supplement face-to-face interview
Work samples	GitHub and Bitbucket Participant websites Client websites	Support or confirm experiences shared during the face-to-face interviews
Work setup	Participant observation	Observe home-based work environment

Table 2. Additional Sources of Data

Data collection in GTM is guided by the emerging theory, i.e., the decisions of which actor in the substantive area to interview next and what questions to ask him or her are influenced largely by which concepts and categories in the emerging theory need further explication and refinement. This sampling process is called *theoretical sampling*.

The other two components of the cyclical process of theory origination are coding via *constant comparison,* and *memoing*, which follow two basic "rules." The first is, "While coding an incident for a category, compare it with the previous incidents in the same and different groups coded in the same category" (Glaser and Strauss 1967, p. 106). The second is, "When you find yourself musing over theoretical notions, stop coding and record a memo of your ideas" (Glaser and Strauss 1967, p. 107). These memos not only serve as a very useful tool for analysis; they would also come in handy when writing up the grounded theory in Phase 3.

Constant comparison produces concepts and categories of increasingly higher abstraction. Typically, several substantive codes (e.g., "Upwork profile", "onlinejobs.ph profile") are combined under or into a broader concept (e.g., "OLM profile"); several concepts (e.g., "OLM profile," "independent portfolios") are combined under a broader category (e.g., "profiling"); several categories (e.g., "profiling," "applying," and "negotiating") are combined under an even more abstract category (e.g., "gig-hunting"); and so on until the highest-level categories that have been produced so far are combined under or into what is called the *core category* (e.g., "gigging"). Therefore, the core category and everything under it is a *grounded* theory of the substantive area. Many substantive grounded theories, including the one in this paper, are processes, though there are many other possible coding families that could be used to guide conceptualization and theory construction (see e.g., Glaser 1978).

The end of Phase 1 is signaled by the emergence of a potential core category (also called core variable) that accounts for most of the variation in the pattern of behavior that has been observed so far. In this study, "transitioning" appeared to be an early potential core category after eight interviews; after the tenth interview, however, it became clear that not all participants viewed their work as merely transitory or temporary. The next potential core category that we saw was "reputationing". However, the 13th participant (and several others who would be interviewed subsequently) was apparently unconcerned about reputation, saying, for example, "I don't know about reputation. However, I do normally go the extra mile for any job or role that I do." Subsequent interviews confirm that, no matter how sexy reputationing sounds, or that two books have recently appeared bearing the title, *The Reputation Economy* (Fertik and Thompson 2015; Gandini 2016), developing one's reputation is neither the participants' main concern nor the core variable that resolves their main concern, though it certainly is part of the core variable. (Indeed, as will be seen in Table 6 later, reputationing turns out to be a strategy for gig-executing.) What eventually emerged as the core category was a cyclical basic social process (BSP) that we called "software gigging," with two main activities, each having three tasks. Several other dimensions of this BSP would emerge in the next phase, theory saturation.

Phase 2: Theory Saturation

In Phase 2, the cyclical process of data collection, coding, and memoing described earlier continues, but this time, coding becomes selective, i.e., limited "to only those variables that relate to the core variable in sufficiently significant ways as to produce a parsimonious theory" (Glaser 2004). The cyclical process ends when theoretical saturation is reached, i.e., when "no additional data are being found whereby the [analyst] can develop properties of the category" (Glaser and Strauss 1967, p. 61). In this study, the final BSP—a cyclical process with two main activities and three stages—emerged after 33 interviews, after which increasingly less and more minor concepts appeared. Nevertheless, we continued to interview 11 more software e-lancers, bringing the total number of participants to 52.

Phase 3: Theory Elucidation

There are three activities in Phase 3 or theory elucidation, and they can be performed in any order or in parallel. First is theoretical coding, which conceptualizes relations among the substantive categories generated in the earlier phases. Second is literature enfolding, in which the extant literature is analyzed to determine how it relates to the substantive theory, and vice versa. Third is theoretical sorting, in which the memos are set up in a theoretical outline in preparation for writing. The results of the these activities will be discussed in the next section.

Other Grounded Theory Methods

There are three major strands of GTM: the Glaserian strand (see e.g., Glaser 1978; Glaser 1998), which is closest to that described in (Glaser and Strauss 1967); the Straussian strand (Strauss and Corbin 1990; Corbin and Strauss 2008), which is more structured (e.g., it uses a coding paradigm involving conditions, context, action/interactional strategies, and consequences), and which some researchers therefore find more comfortable to use than the Glaserian approach; and the constructivist strand (Charmaz 2006) by a student of both Glaser and Strauss in the early 1970s, the decade during which constructivism rose rapidly as an alternative to objectivism.

The choice of which grounded theory method to use, which includes the criteria for judging the resulting grounded theory, ultimately depends on the researcher, especially on the researcher's: (1) need to have special structures (e.g., the coding paradigm of Strauss and Corbin) that can assist in, but unnecessarily constrain, the coding and theory-building process; and (2) ontological and epistemological commitments. Our familiarity with the process of concept induction in artificial intelligence and machine learning (see e.g., Sison 1998), and belief that there are certain realities that exist independently of mental or social constructions, make the Glaserian strand our GTM method of choice.

Results and Discussion

Online E-Lancers' Main Concern and How They Resolve It

Though different software e-lancers might have many, different concerns, their main concern, which is what GTM is designed to discover, can be summed up in two words—*earning flexibly*, i.e., software e-lancers try out and continue doing software e-lancing work because of their desire to earn in a flexible manner. Income is an important driver, but so is flexibility in, say, time and place, as the following sample incidents indicate. (Each sample incident in this paper has a label with two numbers. The first number refers to the participant being quoted. The second number refers to a particular incident in the field notes of a participant's interview.)

> "Flexibility—you can work on your own, with no one instructing you, breathing down you neck. Another is location—like I worked in Dakak (beach resort) for a week." (1.31)
>
> "Flexibility in terms of attire, time when to start work, no "kulba" (anxiety) because I cannot see my boss. I can work anywhere. I cannot miss family events." (4.68)
>
> "It's more of the salary. It is also convenient, especially if you are starting a family." (5.46)
>
> "Income. Average income is 20-30K for a beginner, 40-80K for an expert. In local companies, this is not possible." (9.41)
>
> "I chose this job because I do not want to leave <place>. And the pay is very good." (15.2)
>
> "There are (software development) jobs available locally, like in banks. What is good (about online jobs) is that as long as I'm online, even when, say, I'm with friends, I can work." (18.39)
>
> "It's very flexible. I can go to Boracay (beach resort) and work there. Time is also flexible. Salary is very good." (11.78)
>
> "The compensation, because I cannot earn this amount locally." (22.54)

How to earn in a flexible manner is, therefore, the main concern of these software e-lancers. This is this grounded theory study's first result.

The second result is the substantive theory itself of how the software e-lancers resolve their main concern. This theory takes the form of a basic social process, which we call *software gigging* (Figure 2).

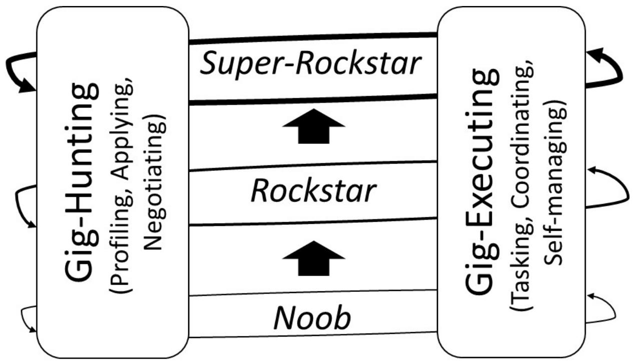
Gig-Hunting
(Profiling, Applying, Negotiating)
Super-Rockstar
Rockstar
Noob
Gig-Executing
(Tasking, Coordinating, Self-managing)

Figure 2. The Basic Social Process of Software Gigging

As Figure 2 shows, the basic social process of software gigging has two dimensions. In the first dimension, software gigging can be viewed as a cyclical process of gig-hunting and gig-executing. Each of these activities have sub-activities or tasks, and strategies to carry out each task. In the second dimension, software gigging can be viewed as progressing in stages, in which e-lancers move up from being noobs, to being rockstars, and finally, super-rockstars.

Software Gigging: The Cycle

Software Gig-Hunting

Gig-hunting is the process of getting a software e-lance job and has three sub-activities or tasks: profiling, applying, and negotiating. (The definitions below of these categories are excerpted from the relevant memos generated during the coding process.)

Profiling is the process of presenting one's skills and qualifications online. Inherent in profiling is the continuous acquisition of skills that are valued in one's target market. There are two major online profiling mechanisms. One is by registering and creating an account profile in an online labor marketplace (OLM) such as Upwork or onlinejobs.ph (e.g., 3.38, Table 3). Another is by creating and posting on one's webpage or on platforms such as GitHub or LinkedIn one's resumé and portfolio (e.g., 24.53, Table 3).

Applying is the process of submitting one's interest in a contracting job (e.g., 25.25, Table 3). Though it may be that the simple act of creating a profile in an OLM can result in a job offer, so that all the e-lancer has to do after profiling is wait, this is not the norm. Instead, an e-lancer might have to send out resumés and cover letters tailored to specific jobs. These documents will, of course, have links to one's online profile.

Negotiating is the process of reaching an agreement in relation to a task, including payment arrangements and other work-related processes. This may or may not involve formal contracts (e.g., 3.17, Table 3).

Table 3 shows sample incidents in the data, how the incidents were coded, the concepts that the codes indicate, and the categories (gig-hunting sub-activities) that the concepts are aspects or components of. As discussed earlier, categories are constructed from concepts, concepts from codes, and codes from the incidents in the data. Reading the table from left to right, one therefore sees the increase in conceptual abstraction brought about by GTM's constant comparison.

Sample Incident in the Data	**Code(s)**	**Concept(s)**	**Category (Sub-Activity)**
"I created an account and profile at oDesk. I constantly add experiences, skills, and other relevant information needed by the platform." (3.38)	OLM Account and Profile	Marketplace Profile	Profiling
"I have my own resumé. I also have a website containing my portfolio." (24.53)	Resumé; Portfolio	Independent Portfolio	Profiling
"I applied. There are many job postings that I applied for." (25.25)	Application	Submitting (Application)	Applying
"I lower my price. For me even if the pay is low, it's okay, as long as there are reviews." (3.39)	Pricing	Bidding	Applying
"No contract. It was just like an informal verbal agreement on how to work, on the salary, and on the schedule."(3.17)	Work Agreement; Salary Agreement; Schedule Agreement	Work Arrangement; Payment Arrangement; Process Arrangement	Negotiating
"I asked for payment upfront to be sure that I will get paid." (36.43)	Payment Agreement	Payment Arrangement	Negotiating

Table 3. Incidents, Codes, Concepts, and Categories of Software Gig-Hunting

Gig-hunting is a competitive process, especially for beginners or what we call *noobs* in the substantive theory. For example:

> "Actually it's hard. Especially if you are a beginner... I need to look at, like, 20 job postings per day. Like, I have limited skills, and things which I do not know. So whoever will respond, I will entertain. At the start, I got 10 to 15 rejections of my application, or no response at all" (9.34)

Due to fierce competition, digital workers seeking to earn opportunities through online labor platforms engage in practices such as underbidding:

"I lower my price. For me even if the pay is low, it's okay, as long as there are reviews." (3.39)

Underbidding in OLMs has been confirmed in quantitative studies such as (Graham, Hjorth, and Lehdonvirta 2017). Noobs and the other stages of software gigging, as well as underbidding and other strategies to carry out the aforementioned tasks, are described in a separate sections.

Software Gig-Executing

Gig-executing is the process of performing and completing a task or project, and has three sub-activities or tasks: tasking, coordinating, and self-managing. (The definitions below of these categories are excerpted from the relevant memos generated during the coding process.)

Tasking is the process of identifying, prioritizing, updating, executing, and distributing the task or project assignment (e.g., 10.17, Table 4), and receiving (or not receiving) payment for the task or project assignment (e.g., 9.10, Table 4).

Coordinating refers to the supporting process of communicating various aspects of the work to the client and team members, if any. It can take different forms that could be as simple as a chat (e.g., 18.14, Table 14), or more complex online work management (e.g., 18.15, Table 4).

Self-managing refers to the process of integrating personal and work boundaries. Self-managing is a result of blurring work boundaries which pertain to coping with task assignments given the setup of work such as working at home (e.g., 28.14, Table 4) or at the beach (e.g., 11.78, Table 4).

Table 4 shows sample incidents in the data, how the incidents were coded, the concepts that the codes indicate, and the categories (gig-executing sub-activities) that the concepts are aspects or components of. As discussed earlier, categories are constructed from concepts, concepts from codes, and codes from the incidents in the data. As in Table 3, reading the table from left to right, one therefore sees the increase in conceptual abstraction brought about by GTM's constant comparison.

Sample Incident in the Data	**Code(s)**	**Concept(s)**	**Category (Sub-Activity)**
"It's like division of labor. When working with a team, I am not assigned to UI, CSS, and styling because I don't accept those kinds of jobs. I'm more on logic, algorithm, that side." (10.17)	Identifying Tasks; Distributing Tasks	Task Execution	Tasking
"When I finished the work, I could no longer contact (my client)." (9.10)	Non-Payment of Work	Task Payment	Tasking
"We use Skype to coordinate." (18.14)	Communication Tools	Communication	Coordinating
"We used a lot of software tools before… also Trello, a tool that will allow us to, like check if I can accomplish it, for the team and the client to know." (18.15)	Project Management Tools	Project Management	Coordinating
"When I was still single, I immediately reported to work (when I woke up). Now, I cook and do laundry while monitoring for work." (28.14)	Work at Home; Flexible Schedule	Interleaving Domestic Task and Work; Flexibility	Self-Managing
"It's very flexible. I can go to Boracay (beach resort) and work there." (11.78)	Work Anywhere	Flexibility	Self-Managing

Table 4. Incidents, Codes, Concepts, and Categories of Software Gig-Executing

Online freelancers commonly working at home experience the blurring of boundaries of personal and work-related concerns. This blurring of boundaries among freelancers is confirmed in studies such as (Cousins and Robey 2015) in its exploration of the role of mobile technologies in managing the boundaries between work and non-work domains in the context of the mobile workforce. The extant literature has two major approaches to examining the blurring of personal and work boundaries: segmentation and integration. Segmentation refers to the "preferences to protect the home domain from work intrusions and to protect the work domain from home intrusions" (Methot and LePine 2016) while integration aims to remove the boundaries between the work and personal domains (Kreiner, Hollensbe, and Sheep 2009).

The three major concepts of tasking, coordinating, and self-managing are interrelated processes: tasks are coordinated, and coordination happens due to tasking around the work environment, which in turn requires self-management. Unlike gig-hunting, which is a linear process, gig-executing is non-linear.

Software Gigging: The Stages

Noob

Noob (short for newbie) refers to beginners, explorers, or those starting to venture into online freelancing, searching for alternative employment or work opportunities online. They tend to consider an online job as part-time or sideline work at first while seeking local employment. As influenced or heard from experienced online freelancers (rockstars), they are the type of online workers who want to try to have extra income or start to have one. This means a noob can also be an experienced company employee in the stage of exploring.

> "As a newbie I am exploring this type of work because I heard that many have gotten rich from onlinejobs." (1.29)
>
> "There is a series of training. I was also required to take English proficiency. After a series of interviews, they hired me, subject to screening, but I was not part of a team. Being new to onlinejobs, I am on probation status, but I have been receiving a minimal salary." (5.10)

The category of noob covers concepts such as "new" and "exploring" (1.29), "beginner" (4.10), and "starting" (17.3). Also, the major strategies for gig-hunting and gig-executing are different from noobs and rockstars. There are also exit indicators for noobs, which means that they will leave online work as a noob when "there are available jobs locally" (18.39), or "they can't handle the time adjustment" (42.24), which is a common requirement of clients. There are cases of online freelancers coming from the industry where they have done considerable software development work. Since they are new to online work, they are considered noobs; however, they might move more quickly to the rockstar stage compared to those for whom software e-lancing is their first job.

Rockstar

Rockstar refers to an experienced online freelance software developer. They will normally have higher rates and salaries. They would also have gained a positive reputation in online labor marketplaces (reviews in Upwork, private reviews in onlinejobs.ph). They are software developers looking for continuous employment online or big projects rather than temporary income source in most of the cases for newbies. They are those who have decided to work full time online from part-time online work, or have acquired good clients and are able to assess and stand for the value of their work.

> "When I first started, I think I lowered my rate. Now I set my rate, then it's up to the client to get me or not. Especially if I can show them my portfolio." (23.27)
>
> "However I believe my colleagues like working with me because, for one thing, they can learn from me. I always give feedback and suggestions to my team; maybe they value those things as well. I have been with teams where they value me more than the team lead/project manager because they can rely on me to provide direction and support when needed." (13.54)

> "I will be handling their clients, like dealing. They want me to manage their company including workers. They are a small entrepreneur. I am usually working with executives. I hold proper meetings even at a distance." (25.73)
>
> "There are also partnerships with a client. Like if they have an idea project for a client, 65% of the payment will be their part, and the rest is mine. If it's not worth the payment, I will not accept it." (36.15)

Rockstars are team leads and are known because of the networks of friends or online developers. In addition, rockstars are also considered an "expert" (9.36), "provide great inputs/direction" (13.55) and give "consultancy" (17.55) to teams or clients. The first indicator that an online freelancer has moved from being a newbie towards a rockstar is the non-negotiable rate. Once they had gained enough experience, they realized that they can still find a client willing to pay the salary rate they think they deserved. In the words of the developer itself, "Usually, we underrated ourselves. I don't know, might not be confident, and it's common to Filipinos. If you lower your rate, the impression of the employer is that you might not be good. I need to balance also, not too cheap or too expensive" (23.77).

Although there are issues with the use of the term rockstar for developers and who they are, there is still a continuous use of the term as people still find it "fashionable" (Tyler 2015).

Super-Rockstar

Super-rockstar refers to a rockstar maximizing the opportunities available by extending work to others and getting a profit from it. It also includes freelancers who have learned the business of their clients and develop their own similar business.

When the online freelancer has a good reputation, it is easier to win a bid or apply for a project. In many cases, they also received job offers. As a result, the practice ofhaving multiple clients is considered by an online freelancer. If they have difficulty handling and coping with client demands, they tend to hire others to help them. The opportunity of increasing income through recruitment, or creating a team is what differentiates the super-rockstars from rockstars.

> "What's the nature of your work? It's mixed, programmer, managing data entry, and also I am the one hiring." (14.7)
>
> "I treat this as my profession and a source of income. As much as possible I will maximize my abilities and opportunities. I'm just lucky that I have many employers." (22. 37)
>
> "If I cannot do the work anymore, I can pass it to others. Like if it's worth 16k I have 3k, they use my name. But some do not have a good performance, so I plan to establish my own team. Like a start-up." (22. 45)
>
> "I had a project before with my client, I replicated it, like marketing. I created my own service." (3.44)

Super-rockstars can be considered as intermediaries in online work. Since they already established online freelancers, they take projects and distribute it to newbies or other online freelancers. The distribution of tasks allows them to finish it even if there are other tasks at hand. In exchange, they usually get a percentage of the income when they pass it to others.

As mentioned earlier (Rockstar section), hiring others indicates that a rockstar is moving into the super-rockstar stage. It is a defining strategy that differentiates a super-rockstar from rockstar. However, hiring a person to help a rock-star is not enough. The defining concept of hiring includes having an intention to get a percentage of the expected income. In others words, profit. It is also apparent that the prevalent strategies in gig-hunting and gig-executing define what it is to be a super-rockstar.

The intermediary role of super-rockstars has also been found in a recent study (Graham et al. 2017) which notes that, "Because of the heavy role that reputational feedback scores play in online gig work platforms, work can flow to intermediaries/middlemen who already have a high score. These intermediaries then re-outsource that work, keeping a part of the client's fee for themselves."

Since being a super-rockstar appears to be the last stage of the online freelancer's professional life, the question is "What's next?" The answer is a registered local business which defines the entrepreneurial aspects of a super-rockstar. This is evident from the changes of the setup of online freelancer software developers when a follow-up interview was done where some of them are already owners of a registered business entity catering to both local and international clients.

Software Gigging Strategies

The software gigging cycle and stages are further differentiated by the strategies that e-lancers use for each gigging activity and gigging stage. For instance, noobs use *potentializing* for gig-hunting, whereas rockstars use *transprofiling*. It will be noted that whereas all software e-lancers can be viewed as going through the gigging cycle of gig-hunting and gig-executing, not all software e-lancers use the strategies we have discovered, and will be describing, in this section.

Gig-Hunting Strategies

Noobs, rockstars, and super-rockstars have different gig-hunting strategies. Table 5 shows sample incidents in the data, the strategy (a category) that the incident indicates, the gig-hunting activity (another category) in which the strategy is used, and the gigging stage (another category) when it is used. The definitions of these strategies are excerpted from the relevant memos generated during the coding process.

Sample Incident in the Data	Category (Gig-Hunting Strategy) that the Incident Indicates	Category (Gig-Hunting Activity) that the Strategy Relates To	Another Category (Gigging Stage) that the Strategy Relates To
"I just keep on applying, like what skills I can possibly do, like 'fake it 'til you make it.'" (2.36)	Potentializing - overstating one's current skills based on an assessment of one's potential capability	Profiling	Noob
"It's really lowering the price. For me even if the payment is low as long as there are reviews." (3.39)	Underbidding - Lowering one's asking price for the purpose of acquiring a job	Applying	Noob
"I also have a website containing my portfolio." (24.53)	Transprofiling - Profiling outside of online labor marketplaces	Profiling	Rockstar
"Now, I set my rate. It's up to the company to get me or not. Especially if I can show my portfolio." (23.27)	Showcasing - A strategy that allows the e-lancer to present his or her portfolio in an impressive manner	Applying; Negotiating	Rockstar
"What is good with Upwork is that I can build an agency profile. Even before it's a team, I can connect and then assign members" (36.27)	Agencying - The presentation of the profile of an online freelancer as a group rather than as an individual.	Profiling	Super-Rockstar
"My client is an agency in the UK, they also outsource here, and they don't know the development side, so they outsource it." (36.14)	Agency Targeting - Identifying and applying to OLM agencies.	Applying	Super-Rockstar

Table 5. Gig-Hunting Strategies and their Sample Incidents and Related Categories

Gig-Executing Strategies

Noobs, rockstars, and super-rockstars also have different strategies for gig-executing. Table 6 shows sample incidents in the data, the strategy (a category) that the incident indicates, the gig-executing activity (another category) in which the strategy is used, and the gigging stage (another category) when it is used. As in Table 5, the definitions of these strategies are excerpted from the relevant memos generated during the coding process.

Sample Incident in the Data	Category (Gig- Executing Strategy) that the Incident Indicates	Category (Gig-Executing Activity) that the Strategy Relates To	Another Category (Gigging Stage) that the Strategy Relates To
"I was working almost 16 hours a day in my first 2 years...Of course, I wanted to impress my clients." (14.20)	Reputationing - Working especially hard so as to impress the client with one's work performance	Tasking; Communicating	Noob
"Through Asana – the structure is like we are a team. I am the full stack developer; one project assigned to me." (1.5)	Collaborating - Working with others to accomplish a project	Coordinating	Rockstar
"Even before it's a team, I can connect [proposals] then assign it to members" (36.27)	Delegating - Dividing one's tasks and assigning these to others (online or offline)	Tasking	Super-Rockstar
"As the lead in one of the 3 projects, I make sure that it is progressing. As manager also, I get to supervise other workers. I am also given the role of determining if a person is capable." (32.46)	Supervising - Managing tasks or a project from recruitment. It includes task distribution and monitoring.	Tasking; Coordinating	Super-Rockstar

Table 6. Gig-Executing Strategies and their Sample Incidents and Related Categories

As mentioned earlier, reputationing, which appears in Table 6 as a strategy for gig-hunting, was an early candidate for the core category. Indeed, one can easily see from an analysis of Tables 5 and 6 that the positive results of reputationing in gig-executing feeds back into gig-hunting (see showcasing, Table 5); in other words, reputationing can actually be viewed as taking place in both the gig-hunting and gig-executing activities of software gigging. However, as mentioned earlier, some e-lancers (e.g., participant 13) are apparently not concerned about their reputation, though they are concerned about doing their best, bringing to mind the oft-quoted John Wooden maxim, "Be more concerned about your character than your reputation" (Wooden and Jamison 1997 p. 28). Moreover, reputation is not the only factor that affects the selection process when gig-hunting; one's rate or price matters as well.

Circumtechventing Strategies

Circumtechventing, from "tech" and "circumvent," is the use of technology to find ways around perceived obstacles to one's main concern. It will be recalled that in the substantive theory of software gigging, the main concern of software e-lancers is *earning flexibly*. In light of this main concern, it would therefore be understandable for software e-lancers to engage in circumtechventing when they perceive the main concern to be too difficult to resolve. In other words, circumventing can be viewed as a way to deal with policies or structures (e.g., platforms) that diminish either earnings or flexibility or both. However, as with all the other strategies previously described in this section, circumventing strategies are not carried out by everyone.

Table 7 shows sample incidents in the data, the circumtechventing strategy (a category) that the incident indicates, the gigging activity (another category) in which the circumtechvention is used, the gigging stage (another category) when it is used, and the aspect of earning flexibly (the main concern), the perceived unnecessary limiting of which might have triggered the circumtechvention. As in Tables 5 and 6, the definitions of these strategies are excerpted from the relevant memos generated during the coding process.

Sample Incident in the Data	**Category (Circumtechventing Strategy) that the Incident Indicates**	**Category (Gigging Activity) that the Strategy Relates To**	**Another Category (Gigging Stage) that the Strategy Relates To**	**Aspect of Main Concern That Might Have Triggered the Circumvention**
"At first, he allowed me to try his account. He told me that if you know it, you can start. First 2 projects, I was using his account." (9.2)	Impostoring - Using someone else's account information to access and perform tasks in an online labor marketplace	Gig-Hunting	Noob	Earnings (It is difficult for noobs, especially with zero experience, to get a gig.)
"Then <first client> said, 'Ok, let's move outside of <OLM>, we can negotiate there.'" (14.62)	Platform Bypassing - Going out of an online platform for client interaction	Gig-Hunting	Rockstar/ Super-Rockstar	Flexibility (Certain OLMs charge what are perceived to be huge fees.)
"My goal is to finish all my tasks. If I cannot do it, I let my siblings help me. They work for me actually. They use my account so it will appear as if I am working on it." (22.17)	Substituting - Allowing others to perform one's task or job on one's behalf	Gig-Executing	Rockstar/ Super-Rockstar	Earning Flexibly (Rockstars get the job done. Does it matter who actually does the job?)
"I have a program that that keeps the screen moving." (22.25)	Tracker Bypassing - Circumventing trackers used to monitor work	Gig-Executing	Rockstar/ Super-Rockstar	Flexibility (Rockstars tend to be output-oriented. Does it matter when or where they do the job?)
"I am also hesitant to demand more, it's because I have 3 jobs and they're practically the same, so I almost am doing nothing, because most of the things are ready." (13.17)	Code Reusing – reusing code written for another, usually previous, client	Gig-Executing	Rockstar/ Super-Rockstar	Earnings (Technology makes it very easy to reuse code. If terms of agreement allow it, why not?)

Table 7. Circumtechventing Strategies and their Sample Incidents and Related Categories

Software e-lancers at the rockstar and super-rockstar levels have no need to engage in impostoring, though their strategy of substituting might encourage noobs to engage in impostoring. Rockstars and super-rockstars also demand, and are often given, greater control (flexibility) over their work. In cases where they are not given the degree of control or flexibility that they believe they deserve, some may use technology to circumvent imposed technological constraints, such as platforms (e.g., 14.62, Table 7) or trackers (e.g., 22.25, Table 7).

Circumtechventing is related to the theory of workarounds (Alter 2014). The said theory can be used to analyze the genesis and evolution of each circumtechventing strategy. What the substantive theory of software gigging contributes are: (1) specific strategies actually used by software e-lancers; and (2) the specific gigging activity—gig-hunting or gig-executing—or constraints on these activities with respect to the main concern of earning flexibly that might prompt a circumtechvention, depending on the level (noob, rockstar/super-rockstar) of the e-lancer.

That circumtechventing has ethical issues should be clear to any reader. For example, when a super-rockstar gets a "substitute," i.e., gets a willing third party to do a particular task, must he or she inform his or her client of this arrangement? Is it not enough for the super-rockstar to deliver on a task? Consequentialism suggests that as long as the client's software requirements are met, it does not matter whether the super-rockstar did the coding himself or herself. As another example, does it matter whether a rockstar reuses code he or she wrote for a previous client, as long as this is within the terms of agreement with his or her clients? (See, for example, the discussion of this in Northcutt and Madden 2004 pp. 396-397.) Space does not permit us to discuss these ethical issues at length. Suffice it to say that the software gigging framework provides conceptual tools for conducting ethical analyses of circumtechventing behaviors.

Evaluating Software Gigging

In *Doing Grounded Theory*, Glaser lists "four criteria for judging and doing grounded theory" (1998, pp. 18-19): fit (does the concept adequately express the pattern in the data which it purports to conceptualize?), workability (do the concepts and the way they are related into hypotheses sufficiently account for how the main concern of participants in the substantive area is continually resolved?), relevance or grab (does the research deal with the main concerns of the participants involved), and modifiability.

The first criterion, validity, nicknamed "fit," is primarily about generating codes from the data, thereby fitting the codes to the data, rather than using categories from the literature and then 'forcing the data' into those categories. Glaser, however, does not forbid getting categories from the literature, as long as the data are not forced to fit such categories. He says, "Since most of the categories of grounded theory are generated directly from the data, the criterion of fit is automatically met." (1978, p. 4) Because all and not just some of the categories in the theory of software gigging were generated from the data, with no categories coming from the literature, the theory has fit.

The second criterion, workability, nicknamed "work," is mainly about ensuring that the theory explains what is going on in the data, i.e., it explains how the actors in the substantive area continually resolve their concern. Glaser notes in *Theoretical Sensitivity* (Glaser 1978, p. 5) that the notion of a basic social process was developed to aid in this effort. So, software gigging explains how software e-lancers earn flexibly using strategies for gig-hunting and gig-executing, depending on what gigging stage they are in.

The third criterion, relevance, nicknamed "grab," is about generating a theory that makes sense to the actors in the substantive area, because the theory is, after all, about their concerns because the idea is that if it makes sense to them, then they might use it. This "align(s) with the pragmatist position on theories and knowledge" (Bryant 2017, p. 79). After the core category in this study had emerged and its salient features were shared with some of the participants, they quickly recognized our constructs.

The fourth criterion, modifiability, is a result of GTM (at least one of its three main strands). Therefore, Glaser says, "the theory is not being verified as in verification studies, and thus never right or wrong… it just gets modified by new data to compare it to" (Glaser 1998, pp. 18-19).

Conclusion

This paper has put forth a substantive grounded theory of online software development freelancing that views *earning flexibly* as the main concern of software e-lancers. To resolve this main concern, they engage in *software gigging*, which is a cyclical process of *gig-hunting* and *gig-executing*, each of which has several sub-activities. Software gigging can also become increasingly lucrative and flexible as one advances from *noob* to *rockstar* to *super-rockstar*. Software e-lancers engage in various strategies such as *reputationing* or *circumtechventing*, as they do software gigging, depending on the specific gigging activity and stage that they are in.

Implications for Practice

The substantive grounded theory of software gigging provides a framework for analyzing not only what software e-lancers do (i.e., the gig-hunting and gig-executing activities and strategies), but also why they do what they do. The answer to the latter has to do with their main concern, which, as stated previously, is *earning flexibly*. So, for example, when a platform or OLM charges fees that workers perceive to be too much, OLMs should expect both clients and e-lancers to do platform bypassing. When clients put too much premium on reputation to the point that no one with no experience ever gets hired, they should not be surprised to hear of impostoring. And when rockstars are treated like noobs so that their every move online is tracked, clients and OLMs should expect all sorts of innovative tracker bypassing mechanisms. The substantive grounded theory of software gigging also provides a roadmap for those just starting out. It shows them what activities are important and what strategies might (or might not) be of help.

Implications for Formal Theory

We surmise that the basic software gigging cycle and stages can also be found in the experiences of software e-lancers in other countries; it is after all logical for project search to precede project execution, and for every expert to begin as a novice. It is the gigging strategies, including the strategies for circumtechvention, that will probably differ from one group or society to another, though there could be strategies or meta-strategies that would be common to many if not all groups. The implication of this is that the basic framework of software gigging (i.e., its cycle and stages) could therefore be used to quickly look for and explore gigging strategies in other groups or societies.

Another implication has to do with other types of gigs. In the previous year, some in our group began exploring, again using GTM, the practices of ride-sharing (e.g., Grab, Uber) drivers/partners. Reflecting now on our work on software e-lancing and on the codes that we have so far generated from data collected from Uber and Grab drivers, it seems that the substantive theory of software gigging might apply, with some modifications, to ride-sharing (driver's side) as well. This suggests that it might be possible for software gigging to be raised to a higher level of abstraction, producing a theory of *gigging*, which could encompass not only online software development freelancing but all manner of Internet-enabled freelance work, such as ride-sharing, medical transcription, and online tutoring.

Acknowledgement

We thank De La Salle University for its STC Pioneer Professor program and the Commission on Higher Education for its scholarship program. We also thank the anonymous reviewers for their encouraging remarks and constructive feedback.

References

Alter, S. 2014. "Theory of Workarounds," Communications of the Association for Information Systems (34:55), pp. 1041–1066.

Bajwa, U., Knorr, L., Di Ruggiero, E., Gastaldo, D., and Zendel, A. 2018. *Towards an understanding of workers' experiences in the global gig economy*. Toronto. Retrieved from: https://www.glomhi.org/gigs

Bryant, A. 2017. Grounded Theory and Grounded Theorizing: Pragmatism in Research Practice, New York: Oxford.

Charmaz, K. 2014. Constructing Grounded Theory: A Practical Guide Through Qualitative Analysis, Thousand Oaks, CA: Sage.

Corbin, J., and Strauss, A. 2008. Basics of Qualitative Research: Techniques and Procedures For Developing Grounded Theory, Thousand Oaks, CA: Sage.

Cousins, K., & Robey, D. 2015. Managing work-lie boundaries with mobile technologies. Information Technology & People 28(1), pp. 34–71.

Cunningham, S. 2015. "Filipinos, Indonesians Top Elance-oDesk Freelancerss in East Asia," *Forbes Asia* (March 13, 2015).

Fertik, M., and Thompson, D. 2015. *The Reputation Economy: How to Optimize Your Digital Footprint in a World Where Your Reputation is Your Most Valuable Asset,* New York: Crown Business.

Gandini, A. 2016. *The Reputation Economy: Understanding Knowledge Work in Digital Society*, London: Palgrave Macmillan.

Garcia, C. 2014. "How Pinoy earned P7.5-M from Freelance Work in Just 1 Year," *ABS-CBN News* (July 27, 2014).

Graham, M., Hjorth, I., and Lehdonvirta, V. 2017. "Digital Labour and Development: Impacts of Global Digital Labour Platforms and the Gig Economy on Worker Livelihoods," *Transfer: European Review of Labour and Research* (23:2), pp. 135–162.

Graham, M., Lehdonvirta, V., Wood, A., Barnard, H., Hjorth, I., and Simon, D. P. 2017. *The Risks and Rewards of Online Gig Work at the Global Margins*, Oxford: Oxford Internet Institute.

Glaser, B. 1978. *Theoretical Sensitivity*, Mill Valley, CA: Sociology Press.

Glaser, B. 1998. *Doing Grounded Theory: Issues and Discussions*, Mill Valley, CA: Sociology Press.

Glaser, B. 2002. *The Grounded Theory Perspective: Conceptualization Contrasted with Description*, Mill Valley, CA: Sociology Press.

Glaser, B. 2004. "Remodeling Grounded Theory," *Qualitative Social Research* (5:2), Art. 4.

Glaser, B., and Strauss, A. 1967. *The Discovery of Grounded Theory*, Chicago: Aldine.

Kässi, O., and Lehdonvirta, V. 2018. "Online Labor Index: Measuring the Online Gig Economy for Policy and Research," *Technological Forecasting and Social Change*, in press.

Kreiner, G., Hollensbe, E., & Sheep, M. 2009. "Balancing Borders and Bridges: Negotiating the Work-Home Interface via Boundary Work Tactics," *Academy of Management Journal* 52(4), pp. 704–730.

Kuek, S. C., Paradi-Guilford, C., Fayomi, T., Imaizumi, S., and Ipeirotis, P. 2015. *The Global Opportunity in Online Outsourcing*, Washington, D.C.: World Bank Group.

Lacity, M., Khan, S., and Yan, A. 2017. "Review of the Empirical Business Services Outsourcing Literature: An Update and Future Directions," in *Outsourcing and Offshoring Business Services*, L. Willcocks, M. Lacity, and C. Sauer (eds.), Cham, Switzerland: Palgrave Macmillan, pp. 499-651.

Methot, J. R., & LePine, J. A. 2016. "Too Close for Comfort? Investigating the Nature and Functioning of Work and Non-work Role Segmentation Preferences," *Journal of Business and Psychology* 31(1), 103–123.

Northcutt, S., and Madden, C. 2004. *IT Ethics Handbook: Right and Wrong for IT Professionals*, Rockland, MA: Syngress.

Orlikowski, W. 1993. "CASE tools as organizational change: investigating incremental and radical changes in systems development," *MIS Quarterly* (17), pp. 309-340.

Simmons, O. 2011. "Why classic grounded theory? In *Grounded Theory: The Philosophy, Method, and Work of Barney Glaser*, V. Martin, and A. Gynnild (eds.), Boca Raton, FL: Brown Walker, pp. 15-30.

Sison, R. 1998. *Multistrategy Detection and Discovery of Novice Programmer Errors* (Doctoral Dissertation), Department of Computer Science, Tokyo Institute of Technology, Tokyo, Japan.

Sison, R. 2017. *Simulchieving Forward: A Grounded Theory of Presidential Leadership of Highly Effective Higher Education Institutions* (Doctoral Dissertation), Department of Educational Leadership and Management, College of Education, De La Salle University, Manila, Philippines.

Strauss, A., and Corbin, J. 1990. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*, London: Sage.

Tyler, J. 2015. *Building Great Software Engineering Teams*. APress.

Urquhart, C., Lehmann, H., and Myers, M. 2010. "Putting the 'Theory' Back into Grounded Theory: Guidelines for Grounded Theory Studies in Information Systems," *Information Systems Journal* (20), pp. 357-381.

Urquhart, C., and Fernandez, W. 2013. "Using Grounded Theory Method in Information Systems: The Researcher as Blank Slate and Other Myths," *Journal of Information Technology* (28), pp. 224-236.

Wooden, J., and Jamison, S. 1997. *Wooden: A Lifetime of Observations and Reflections On and Off the Court*, Chicago: Contemporary Books.