

# Agile Undercover: When Customers Don't Collaborate

(author 1), (author 2), and (author 3)

(address),  
(blinded for review)  
{Email@blindedforreview}  
(url)

**Abstract.** *Customer collaboration is vital to Agile projects. Through a Grounded Theory study of New Zealand and Indian Agile teams we discovered that lack of customer involvement was causing problems in gathering and clarifying requirements, loss of productivity, and business loss. "Agile Undercover" allows development teams to practice Agile despite insufficient or ineffective customer involvement. We present the causes and consequences of lack of customer involvement on Agile projects and describe the Agile Undercover strategies used to overcome them.*

**Key words:** Agile Software Development, Customer Involvement, Agile Undercover, Grounded Theory

## 1 Introduction

Customer involvement in traditional software development projects is typically limited to providing the requirements in the beginning and feedback towards the end, with limited regular interactions between the customer and the development team [15, 16, 20, 27]. In contrast, customer collaboration is a vital feature and an important success factor in Agile software development [5, 8, 17, 18, 25, 26, 27]. Agile methods expand the customer role within the entire development process by involving them in writing user stories, discussing product features, prioritizing the feature lists, and providing rapid feedback to the development team on a regular basis [9, 15, 25, 27].

In this paper, we present the results of a Grounded Theory study, involving 30 Agile practitioners from 16 different software development organizations in New Zealand and India. Our study revealed that *Lack of Customer Involvement* was one of the biggest challenges they faced. We analyze the causes and consequences of lack of customer involvement and present *Agile Undercover* — a set of strategies used by Agile practitioners to overcome the lack of customer involvement on Agile projects [19]. The rest of the paper is structured as follows: section 2 describes our research method followed by the results of the study in sections 3 to 9. Section 10 is discussion of our findings in light of related works. Section 11 describes limitations of our study followed by the conclusion in section 12.

## 2 Research Method

Grounded Theory (GT) is the systematic generation of theory from data analyzed by a rigorous research method [12, 13]. GT was developed by sociologists Glaser and Strauss [14]. We chose GT as our research method for several reasons. Firstly, Agile methods focus on people and interactions and GT, used as a qualitative research method, allows us to study social interactions and behaviour. Secondly, GT is most suited to areas of research which have not been explored in great detail before, and the research literature on Agile team-customer relationships is scarce [15]. Finally, GT is being increasingly used to study Agile teams [6, 7, 25, 31]. Following Glaser's guidelines, we started out with a general area of interest — Agile project management — rather than beginning with a specific research problem [7].

### 2.1 Data Collection

We collected data by conducting face-to-face, semi-structured interviews with Agile practitioners using open-ended questions. The interviews were approximately an hour long and focused on the participants' experiences of working with Agile methods, in particular the challenges faced in Agile projects and the strategies used to overcome them. We also observed several Agile practices such as daily stand-up meetings (co-located and distributed), release planning, iteration planning, and demonstrations. In order to get a rounded perspective, we interviewed practitioners in various roles: Developers, Agile Coach (Scrum Master and XP Coach), Agile Trainer, Customer, Business Analyst, Tester, and Senior Management. Data collection and analysis were iterative so that constant comparison of data helped guide future interviews and the analysis of interviews and observations fed back into the emerging results.

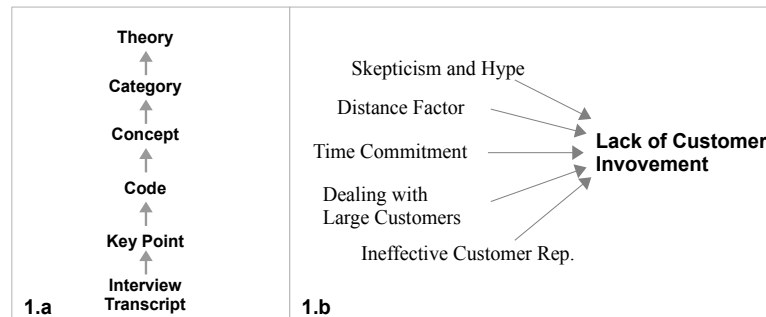
### 2.2 Data Analysis

We used open coding to analyze the interview transcripts in detail [12, 10]. We began by collating key points from each interview transcript [10]. Then we assigned a *code* — a phrase that summarizes the key point in 2 or 3 words — to each key point [1]. The codes arising out of each interview were constantly compared against the codes from the same interview, and those from other interviews and observations. This is GT's *constant comparison method* [11, 14] which was used again to group these codes to produce a higher level of abstraction, called *concepts* in GT.

The constant comparison method was repeated on the concepts to produce another level of abstraction called a *category*. As a result of this analysis, the concepts *Skepticism and Hype*, *Distance Factor*, *Lack of Time Commitment*, *Dealing with Large Customers*, and *Ineffective Customer Representative* gave rise to the category *Lack of Customer Involvement*. These concepts help describe the category *Lack of Customer Involvement* and are referred to as its *properties* [11].

Another set of concepts uncovered from the analysis include *Changing Priority*, *Risk Assessment Up-Front*, *Story Owners*, *Customer Proxy*, *Just Demos*, *E-collaboration* and *Extreme Undercover*. These concepts led to the emergence of the category *Agile Undercover*.

Fig 1.a shows the levels of data abstraction using GT and Fig 1.b illustrates how the category *Lack of Customer Involvement* emerged from underlying concepts.



**Fig. 1.** a: Levels of data abstraction in GT b: Emergence of category *Lack of Customer Involvement* from concepts

We analyzed the observations and compared them to the concepts derived from the interviews. We found our observations did not contradict but rather supported the data provided in interviews, thereby strengthening the interview data. The use of *memoing* — theorizing write-up of ideas about codes and their relationships — was vital in recording the relationships between codes [12]. The conceptual *sorting* of memos was done to derive an outline of the emergent theory, showing relationships between concepts.

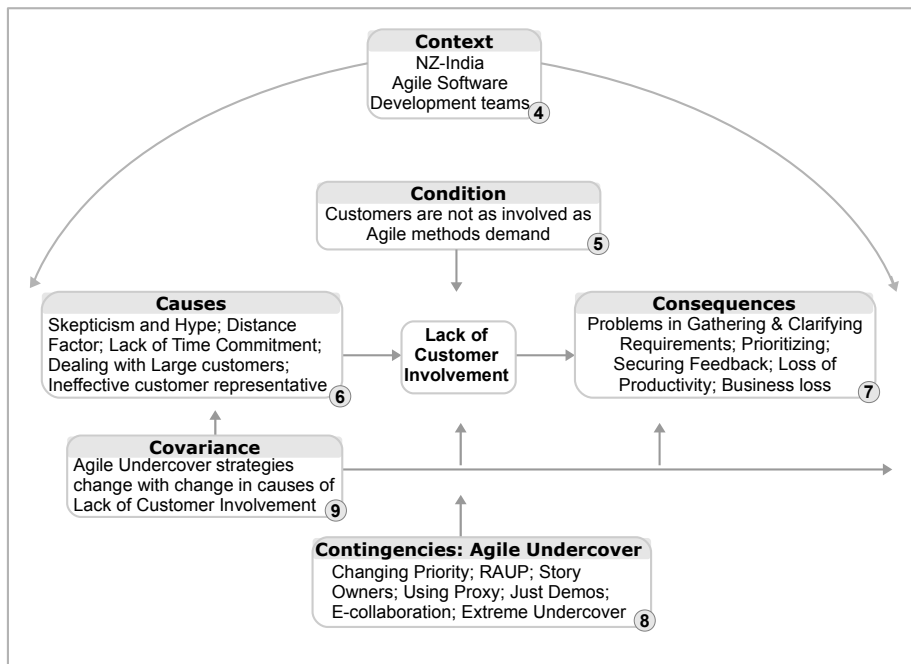
### 2.3 Generating a Theory

The final step of GT is generating a theory, also known as *theoretical coding*. Theoretical coding involves conceptualizing how the categories (and their properties) relate to each other as a hypotheses to be integrated into a theory [12]. Following Glaser's recommendation, we employed theoretical coding at the later stages of analysis [11], rather than being enforced as a coding paradigm from the beginning as advocated by Strauss [30].

Glaser lists several common structures of theories known as *theoretical coding families* [12, 13]. By comparing our data with the theoretical coding families, it emerged that the coding family best 'fit' for our data was the *Six C's* coding family [12, 13, 21]: Contexts, Conditions, Causes, Consequences, Contingencies, and Covariances. Using the *Six C's* theoretical model we describe (1) Contexts: the ambiance (Agile development teams in NZ and India) (2) Conditions: factors that are prerequisites for the category, *Lack of customer involvement*, to manifest (3) Causes: reasons that cause lack of customer involvement (4) Consequences: outcomes or results of lack of customer involvement (5) Contingencies: moderating factors between causes and consequences (*Agile Undercover* strategies) (6) Covariances: correlations between different categories (*Agile Undercover* strategies change when factors that cause *Lack of Customer Involvement* change).

### 3 Results

In the following sections we present our theory. We have adapted Glaser's Six C's model diagram [12] to illustrate our theory of lack of customer involvement (Figure 2). The category *Lack of Customer Involvement* is at the center of the diagram. Each of the Six C's are represented in the other rectangles in relation to the central category, with corresponding section numbers (in circles) where we describe them.



**Fig. 2.** Six C's as properties of category *Lack of Customer Involvement*

In the following sections, we have selected quotations drawn from our interviews that shed particular light on the concepts. Due to space reasons we cannot describe *all* the underlying key points, codes, and concepts from our interviews and observation that further ground the discussion.

### 4 Context

We interviewed 30 Agile practitioners from 16 different software development organizations over 2 years, half of whom were from New Zealand and half from India. Figure 3 shows the participants and project details. In order to respect their confidentiality, we refer to the participants by numbers P1 to P30. All the teams were using Agile methods, primarily combinations of Scrum and eXtreme Programming (XP) —

two of the most popular Agile methods today [3, 28, 29]. The teams practiced Agile practices such as iterative development, daily stand-ups, release and iteration planning, test driven-development (TDD), continuous integration and others. Participants' organizations offered products and services such as web-based applications, front and back-office applications, and local and off-shored software development services.

Participants	Agile Position	Agile Method	Org. Size	Country	Domain	Team Size	Project Duration	Iteration
P1	SM	Scrum & XP	S	NZ	E-commerce	4	2	4
P2-P4	Dev × 3	Scrum & XP	S	NZ	Environment	4 to 6	12	1
P5	AC	Scrum & XP	L	NZ	Social Services	4 to 10	3 to 12	2
P6	Cust Rep	Scrum	XS	NZ	Entertainment	6 to 8	9	4
P7-P12	AC, BA, Tester, Dev × 2, Cust Rep	Scrum	M	NZ	Health	7	9	2
P13	AC	Scrum & XP	XL	NZ	Telecom & Transportation	6 to 15	12	4
P14	AC	Scrum & XP	S	NZ	Government Education	4 to 9	4	2
P15	Dev	Scrum & XP	XS	NZ	Software Development	7	6	2
P16-P20	Dev × 3, SM × 2	Scrum & XP	S	India	Software Development & Consultancy	5	6	2
P21	AC	Scrum & XP	L	India	Telecom	8 to 15	3	4
P22-P25	AC × 4	Scrum & XP	M	India	Software Development	7 to 8	3 to 6	2
P26	AC	Scrum & XP	S	India	IT & Agile Training	7 to 8	48	3
P27	Designer	Scrum & XP	S	India	Web-based services	5	1	2
P28	AC	Scrum & XP	M	India	Financial Services	8 to 11	36	2
P29	AT	Scrum	XS	India	Agile Training	7	8	3
P30	BA	Scrum & XP	M	India	Software Development	15	12	1

**Fig. 3.** Participant and Project Contexts. (Agile Position: Agile Coach (AC), Developer (Dev), Customer Rep (Cust Rep), Business Analyst (BA), Senior Management (SM), Agile Trainer (AT); Organizational Size: XS < 50, S < 500, M < 5000, L < 50,000, XL > 100,000 employees; Project duration is in months & iterations are in weeks.)

The level of Agile experience varied across the different teams. While some teams had under a year of experience, others had been practicing Agile for over 5 years. The Indian teams were mostly catering to off-shored customers in Europe and USA and most of the NZ teams were catering to in-house customers, some of whom were located in separate cities. We include more details of the context in sections below as necessary.

## 5 Condition

Most participants did not receive the level of customer involvement that Agile methods demand (P1-P12, P14-P19, P21-P23, P25, P26, P28-P30). Lack of customer involved was seen as “*the most difficult part of Agile*” and “*the biggest problem*” because “*Agile [requires] fairly strong customer involvement*” (P4, P17, P30).

## 6 Causes

### 6.1 Skepticism and Hype

Customers harbour skepticism about Agile methods and pose resistance to involvement. They don't readily understand Agile practices such as 'fail fast' and its intended benefits — that minimum time and effort may be wasted on a project that's bound to fail — and become prejudiced:

*“Agile terms...say option to fail early - believe me customers don't want to hear it. Customers don't want to admit that there could be some problem with the nice idea they put on paper. Forget about fail early, we dont want to fail at all!”*  
— P19, Senior Management, India

On the other extreme, with the increasing popularity of Agile methods, some customers treat Agile as a buzzword and are eager to reap the benefits of Agile projects without fully understanding their own responsibilities of collaboration and involvement:

*“I mostly work [with] Indian companies with client in US...they see is client can make changes all the time and they think wow that sounds great!...They don't understand the counter-balancing discipline...customer involvement.”* — P29, Agile Trainer, India

Agile practitioners mentioned both skepticism and hype as a major cause for lack of customer involvement (P5, P11, P14, P16, P17, P19, P22, P26, P29).

### 6.2 The Distance Factor

The majority of the Indian participants were catering to off-shored customers in Europe and USA which made customer collaboration challenging for them due to geographic and time-zone differences.

*“Customer involvement is poor, very adversarial relationship. Basically the customers big fear is being cheated — because they are far away, they don't know the team — every mistake seems like an indication of incompetence or vendors trying to deceive [them].”* — P29, Agile Trainer, India

The effect of distance was apparent on a NZ team whose local customer was actively involved while their distanced customer was unwilling to participate.

*“Relationship with [distanced customer] is very different from one in [same city]. We can call at short notice...and can we borrow somebody for half a day, they are willing to do it.”* — P9, Tester, NZ

Distance between the team and their customers promoted misunderstandings (P11) and caused lack of customer involvement due to problems of communicating and coordinating over distances [P8-P12, P17, P29).

### 6.3 Lack of Time Commitment

Teams complain of not receiving enough collaboration time from their customer reps:

*"[Customers say] 'I want to have Taj Mahal' but of granite or marble? They don't even have time to talk about that!"* — P25, Agile Coach, India.

At the same time, they realize that the customer rep's operational job may sometimes take precedence over their involvement on Agile projects because the rep has "*got a full-time job...and is quite busy, trying to fit us in.*" (P7). Development teams find the ability of the customer rep to devote time for collaboration is dependent on the customer (boss).

*"I've never worked on [a project] where customer representative was given enough time to really be able to do the amount that they should."* — P2, Developer, NZ

Eliciting adequate time from customers was challenging for the Agile teams and a cause for lack of customer involvement (P2, P4, P5, P7, P15, P19, P25, P26).

### 6.4 Dealing with Large Customers

Large customers and customers with larger projects showed a preference for traditional ways of working and were unwilling to collaborate as Agile customers.

*"Larger the organization, they are a bit less flexible towards trying out new things."* — P16, Developer, India

Large customers were often not bothered about the internal development-level practices of smaller Agile vendor companies, and tried to assert their traditional style of working on the team:

*"Because of their size they were running hundreds of projects, they didn't want to care that this small organization was talking about, they just wanted to have things done in their own way."* — P17, Developer, India

We found evidence of this only in India (P16, P17) and not in New Zealand.

### 6.5 Ineffective Customer Representative

While Indian Agile teams had limited face-to-face interactions with their customers, some NZ teams had a customer employee assigned to the project as a *customer rep*. An effective customer rep was described as "*someone who understands the implications of that system...where it fits into the business process*" and at the very least "*someone who knows how to use a computer!*" (P5, P9). Some NZ practitioners found their respective customer reps to be ineffective in providing timely requirements and feedback, while others found them lacking in proper understanding of Agile practices.

*“Unfortunately the person who is [the customer rep] has an I.Q. of literally 25...doesn't really know how the current system works, doesn't know much about the business process, is petrified of the project sponsor, and is basically budget-driven. So she doesn't really care if it's not going to work in a way that the end users like.”* (undisclosed) Developer

Several NZ participants expressed their frustration over not being able to choose the customer reps (P2, P7, P8, P9). It is not enough to have a customer rep for the project, it is also important for that rep to be effective in providing requirements and feedback to the team. An ineffective customer rep was a contributing factor to the Agile teams wanting more or better customer collaboration.

## 7 Consequences

### 7.1 Problems in Gathering and Clarifying Requirements

Agile teams found gathering requirements from customers as *“one of the worst things”* and *“biggest frustration”* on the projects (P8,P10). Getting customer reps to clarify requirements are also a problem because of their unavailability:

*“Things [awaiting clarification] would queue up for them and then they'd just answer the whole queue at once...then as soon as they got busy again it would start to get a bit harder.”* — P2, Developer, NZ

As a result of insufficient or ineffective customer involvement, the development teams were unable to gather requirements and to get customer reps to clarify them in time for development to commence.

### 7.2 Problems in Prioritizing Requirements

Providing and clarifying requirements is not enough, the customers are also required to prioritize them in order of business value. Understanding and using the concept of prioritization doesn't always come naturally to customers:

*“We're meant to have one list of product backlog and it's supposed to be prioritized but when the client says ‘Oh that's all priority’ we have to go back and say ‘which?! what do you mean?!...you can't have all priority!’”*. — P11, Developer, NZ

*“Customer needs to tell his priorities that this is the first thing we want.”* — P16, Developer, India

Teams faced difficulties in getting customer reps to prioritize the requirements and as such they were unsure about what features to develop and deliver first. (P7, P16)



### 7.3 Problems in Securing Feedback

Customer feedback is of vital importance in ensuring the desired product is being developed and delivered incrementally. As a senior developer pointed out “*the whole point of the two week iterations was so that the end users could know if we were on the right track*” (P15) and required customer reps providing feedback on developed features.

*“If [the customer reps] didn’t respond you just didn’t care about their opinion...and at the end of the project...the business units that didn’t give much feedback, when it went to a user, started complaining. And it’s like well if we didn’t get any critique it’s not really our fault!”* — P2, Developer, NZ

In absence of customer feedback, teams were unable to assess how well the features met the requirements.

### 7.4 Loss of Productivity

Inability to gather requirements in time for the iterations could result in “*the project get[ting] stalled*” (P5) or loss of productivity:

*“The team has the capacity...[but] with Agile if you don’t have the requirement you can’t do anything...because you are supposed to be in-line with business.”* — P10, Developer, NZ

Without clear requirements and feedback, the teams were forced to “*make more business decisions than [the team] would like*” (P15) and as a result would get “*misaligned from the desired business drivers*” (P5) consequently requiring costly rework (P2, P4, P15). Rework is taxing for developers because they have to revisit stories developed several iterations ago due to delay in customer feedback:

*“Yes [we had to rework] but it’s not the re-work, it’s re-worked easily as long as it’s near the time you did it. So having to go back and augment what you did three weeks ago was [hard].”* — P2, Developer, NZ

### 7.5 Business Loss

The most extreme consequence of lack of customer involvement was business loss that the vendor organization suffered because there was “*no match between what Agile says and the way [the customers] wanted.*” (P17). Some customers were explicitly opposed to Agile practices and did not want to be involved in an Agile project. Some Agile vendor organizations, in such cases, decided to suffer business loss over working on an Agile project with no customer involvement.

*“We’ve lost business multiple times...We try to find out early on if [Agile is] gonna be a problem and if it is, we say ‘Okay, lets go our separate ways.’”* P1, Senior Management, NZ

## 8 Contingencies: Agile Undercover

### 8.1 Changing Priority

In an effort to maintain the iterative and incremental nature of their Agile projects, teams were forced to change priority of user stories that were awaiting customer requirements, clarification, or prioritization. Such stories were usually demoted in priority and pushed further down into the product backlog until the required customer response was secured and development on those stories could re-commence. Agile teams confessed to that they changed the priority of the story in absence of enough, clear, and prompt requirements (P2, P8, P14, P26).

*“[If] we know exactly what business want or we know eighty percent of what they want, we include that story in the sprint; otherwise if we have something that’s a little bit unsure, we don’t include that in the sprint.”* — P10, Developer, NZ

A similar strategy, called *definition of ready* [2] was adopted by an Indian team:

*“We have recently started using...the definition of ready....product owner will not take something that is not ‘done’ and similarly developers are not going to take something that’s not ‘ready’.”* — P18, Developer, India

A user story was considered *ready* when the customers had provided the business goals and expected outcome associated with the story and implementation details necessary to estimate the story had been discussed. A story that was not *ready* was not able to achieve priority in the product backlog.

### 8.2 Risk Assessment Up Front

One of the participants mentioned using an Agile risk assessment questionnaire as a basis to gauge the level of customer involvement on the project up front. The questionnaire included questions such as ‘what is the commitment of the customer rep in terms of time?’ and had multi-choice answers:

*“‘They’re either assigned to this project or available as a first priority’ [is] the best situation and the worst situation [is] ‘Just as time allows’”* — P14, Senior Agile Coach, NZ

Performing risk assessment upfront before the start of the project allows the team to discover if the indicated level of customer involvement is a potential risk to the Agile project. Usually the reason behind limited involvement is lack of funding or unavailability of the customer rep. The approach taken to overcome this problem was to negotiate with the customer for freeing up the customer rep’s time by providing funding from the project. The aim was to *“allow [the rep] to become a project team member, in a more permanent way for the duration of the project”* (P14) which is the ideal level of customer collaboration on Agile projects [3, 8, 29]

### 8.3 Story Owners

The practice of assigning story owners was an adaptation to the Scrum practice of allocating a product owner [29]. Story owners were responsible for particular stories (less than a week long), instead of *all* the stories in the product backlog: *“every story had to have an owner to get into prioritisation.”* (P14) Assigning story owners served a three-fold purpose. Firstly, having multiple story owners instead of a single customer rep for entire project meant no one person from the customer's organization was expected to be continuously available.

*“We didn't need that story owner for the duration of the project, we normally only need them for part of an iteration.”* – P14, Senior Agile Coach, NZ

Secondly, it allowed the team to plan out stories for development in synchronization with the corresponding story-owner's availability. Thirdly, it encouraged a sense of ownership among customer reps as they were encouraged to present their own stories to peers at end of iteration reviews.

*“We get the [story owners] to demonstrate those stories to their peers at the end of the iteration review, this concept is something we've evolved over the project.”* — P14, Senior Agile Coach, NZ

After one such presentation a particularly skeptical customer rep was *“quite chuffed [pleased], and at the [next] iteration planning meeting, that person was all go! Instead of sitting back with their arms folded, they had their elbows on the table, leaning forward, and were driving the story detailing conversations we were having.”* (P14)

### 8.4 Customer Proxy

Some Agile teams used a customer proxy — a member of the development team coordinating with the customers — to secure requirements and feedback. The use of proxy was visible in Indian teams where the customers were physically distant (P16, P19, P23, P28).

*“Using Client proxy, so we assign a customer representative who interacts with the team much often but then passes on the feedback from the customer to the team and vice versa.”* — P28, Agile Coach, India

The use of proxy to co-ordinate between the customers and the team was also observed in New Zealand, where a Business Analyst and couple of developers on different teams served as the proxies because of their communication skills (P2, P4, P8).

*“We've got two people [playing proxy]...[due to] their ability to communicate ideas; they're well-spoken and able to get those ideas across...which is great for developers!”* — P3, Developer, NZ

### 8.5 Just Demos

Despite their reluctance or inability to attend other meetings, almost all customers were interested enough to attend demonstrations (demos) as it gave them an opportunity to see new functionalities of their software (P7, P11, P17). Demos were often the *only* regular collaboration that these Agile teams received from the customer reps and they used this opportunity to discuss features and receive feedback. As the local and involved customer rep of a NZ team disclosed:

*“Just the sprint demos...and [we see] three pieces of functionality and it's all done in fifteen minutes, we take the full hour to discuss the other things...the demos were fun. I don't know if that's their intent, but they were!”* — P12, Customer Rep, NZ

### 8.6 E-collaboration

Electronic collaboration (e-collaboration) was a popular means of regularly communicating with customers using video/voice conferencing, phone, email and chat. For Indian teams with off-shored customers, e-collaboration was a practical work-around:

*“Video conferencing becomes very important. Its all about collaboration [when] time difference is a problem...with Europe [there is a] 4 hours overlap.”* — P17, Developer, India

Teams used web-conferencing and chats to conduct stand-up meetings and demos over the web (P13). New Zealand teams were also observed using phone conferencing with shared documents and emails using online forums called webEx (P9) and Skype, which *“doesn't cost that much to use; Skype [costs] literally zero.”* (P10) E-collaboration was particularly important for our Agile teams because (a) Agile requires *regular* customer involvement (b) several teams had physically distant customers making face-to-face collaboration difficult and (c) e-collaboration was a convenient and inexpensive.

### 8.7 Extreme Undercover

In an effort to avoid extreme consequences of lack of customer involvement such as business loss, Agile teams chose to follow Agile practices internally at the team level while keeping the customer unaware:

*“In none of the [three] cases the customer was aware of Agile, they didn't really want to do Agile...but what we had done was...taken charge of the projects [and] we had made it Agile - internally following Agile.”* — P17, Developer, India

Other practitioners confessed that they *“don't mention the ‘A’ word”* to customers who were explicitly opposed to Agile despite all the team's efforts to convince them (P28). Over the span of the research study, however, we have observed a decrease in the use of this particular practice due to increasing popularity of Agile methods among customers.

## 9 Covariance

Covariance occurs when one category changes with the changes in another category [12, 21]. We found that *Agile Undercover* strategies vary with the factors that cause *Lack of Customer Involvement*. For example, *Customer Proxy*, *Just Demos* and/or *Changing Mindset*<sup>1</sup> were found in practice where participants faced *Ineffective Customer Reps*. Figure 4 presents the covariance relationships between *Agile Undercover* strategies being used and the factors that cause *Lack of Customer Involvement*.

	Changing Priority	RAUP	Story Owners	Customer Proxy	Just Demos	E-Collab	Extreme Undercover	Changing Mindset <sup>1</sup>
Skepticism & Hype			✓		✓		✓	✓
Distance Factor				✓	✓	✓		
Time Commitment	✓	✓	✓		✓			
Large Customers							✓	✓
Ineffective Customer Rep				✓	✓			

Fig. 4. Covariance between Agile Undercover and Lack of Customer Involvement. RAUP is Risk Assessment Up-Front.

## 10 Discussion and Related Work

While some *Agile Undercover* strategies such as *Changing Priority*, *Risk Assessment Up-front*, *Story Owners* and *Extreme Undercover* adapted existing practices, others such as using *Customer Proxy*, *Just Demos* and *E-collaboration* were existing Agile practices used specifically to overcome lack of customer involvement. Following classic GT, we discuss implications of *Agile Undercover* in light of existing literature after presenting the our research results. [14].

Using the *definition of ready* for user stories [2] forced customers to provide detailed requirements with clear business drivers. The *definition of ready* complemented the existing Scrum definition of done [29]. The practice of assigning *Story Owners* was an adaptation of the existing product owner practice. Unlike the product owner [8, 29], the story owner was only responsible for one story at a time. This was an effective way of overcoming the limited availability of customer reps. Story owners also provide an alternative to the practice of on-site customer which has been found to be effective but burdening and unsustainable for long-term use [8, 15, 22, 25].

A *Customer Proxy* is known to be used in situations where customer involvement is not ideal [20, 23, 24]. Participants agreed that being a proxy was demanding yet useful in co-ordinating with distant customers (P10, P11, P16, P28). Face-to-face communication is considered “*the most efficient and effective method of conveying information to and within a development*” [20, 17], followed by video-conferencing, telephone, and email [22]. Our participant used *E-collaboration* extensively but noted that “*it does not take the place of having somebody sitting beside you*” (P9). Other limitations were imposed by the tool itself, such as Skype not supporting three or more people through

<sup>1</sup> an *Agile Undercover* strategy used to convince customers by highlighting benefits of Agile methods — described earlier [18] and not reiterated here for space reasons.

video chatting (P10). Although demos are a regular Agile feature, they were often the only face-to-face collaboration time our participants received from their customers and they used *Just Demos* to discuss features and receive clarifications in addition to feedback.

The customer rep is ideally an individual who has both thorough understanding of and ability to express the project requirements and the authority to take strategic decisions [9, 15, 27]. Boehm advocates dedicated and co-located CRACK (Collaborative, Responsible, Authorized, Committed, Knowledgeable) customers for Agile projects [4]. In addition, our participants suggested that customers should choose reps that understand Agile practices and their own responsibilities in the process of Agile software development (P5, P12, P29).

*Extreme Undercover* was used by Indian teams as a last resort, specially when facing business loss. While one NZ team disclosed facing business loss, they chose to bear it. Over the research period (2 years) we found that the use of *Extreme Undercover* diminished with the increase in popularity of Agile methods. Participants found that *Agile Undercover* strategies were effective in ensuring the success of their projects despite insufficient or ineffective customer involvement.

## 11 Limitations

Since the codes, concepts, and category emerged directly from the data, which in turn was collected directly from real world, the results are grounded in the context of the data [1]. We do not claim the results to be universally applicable: rather, they accurately characterize the context studied [1]. Our choice of research destinations and participants were limited in some ways by our access to them.

## 12 Conclusion

We conducted a GT study, involving 30 Agile practitioners from 16 different software development organizations in New Zealand and India, over a period of 2 years. The results reveal that customers are not as involved on these Agile projects as Agile methods demand. In this paper, we have described (a) the causes of lack of customer involvement (b) its the adverse consequences on Agile projects and (c) *Agile Undercover* strategies used by our participants to practice Agile despite insufficient or ineffective customer involvement. Some *Agile Undercover* strategies were adapted practices while others were close to existing Agile practices. Although we do not prescribe *Agile Undercover* strategies as replacement for real and valuable customer involvement, they may assist Agile teams facing similar lack of customer involvement. Participants found the *Agile Undercover* strategies to be largely useful and effective in their own contexts. Future studies could explore the viability and success of these strategies in different contexts such as in other countries and cultures.

**Acknowledgments.** Our thanks to all the participants. This research is generously supported by (blinded for review) grant and a (blinded for review) scholarship.

## References

1. Adolph, S. et al.: A Methodological Leg to Stand on. In CASCON, ACM, USA (2008)
2. Beaumont, S.: The Definition of READY. Xebia blogs. DOI=<http://blog.xebia.com/2009/06/19/the-definition-of-ready> (4 Dec 2009)
3. Beck, K.: Extreme Programming Explained: 2nd Edition, Addison-Wesley (2004)
4. Boehm, B., Turner, R.: Rebalancing Your Organization's Agility and Discipline. XP (2003)
5. Chow, T., Cao, D.: A survey study of critical success factors in agile software projects. *J. Syst. Softw.*, pp.961–971 (2008)
6. Cockburn, A.: People and Methodologies in Software Development. PhD thesis, University of Oslo, Norway, (2003)
7. Coleman, G. et al.: Using GT to Understand Software Process Improvement: A Study of Irish Software Product Companies. *J. Inf. Softw. Technol.*, 49(6), pp.654–667 (2007)
8. Dybå, T., Dingsoyr, T.: Empirical Studies of Agile Software Development: A Systematic Review. *J. Inf. Softw. Technol.*, 50(9-10), pp.833–859 (2008)
9. Fraser, S. et al.: The Role of the Customer in Software Development: the XP Customer - Fad or Fashion? In OOPSLA, pp. 148–150, ACM, USA (2004)
10. Georgieva, S., Allan, G.: Best Practices in Project Management Through a Grounded Theory Lens. *E. J. Business Research Methods* (2008)
11. Glaser, B.: Basics of Grounded Theory Analysis: Emergence vs. Forcing. CA (1992)
12. Glaser, B.: Theoretical Sensitivity. Sociology Press, Mill Valley, California (1978)
13. Glaser, B.: The Grounded Theory Perspective III: Theoretical Coding. Sociology Press, Mill Valley, California (2005)
14. Glaser, B., Strauss, A.L.: The Discovery of Grounded Theory. Aldine, Chicago (1967)
15. Grisham, P. S., Perry, D. E.: Customer relationships and Extreme Programming. In HSSE '05, ACM, USA (2005)
16. Hanssen, G.K., Fgri, T.E.: Agile Customer Engagement: A Longitudinal Qualitative Case Study. In: ISESE. pp. 164–173, Brazil (2006)
17. Highsmith, J., Fowler, M.: The Agile Manifesto. *Software Development Magazine* (2001)
18. Blinded for review.
19. Preliminary results presented orally at work-in-progress session of blinded-for-review conference, no publication in proceedings.
20. Judy, K.H., Krumins-Beens, I.: Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership. In HICSS. pp.462–462, Hawai (2008)
21. Kan, M.M., Parry, K.W.: Identifying Paradox: A grounded theory of leadership in overcoming resistance to change. *The Leadership Quarterly*, pp.467–491 (2004)
22. Korkala, M., Abrahamsson, P., and Kyllonen, P.: A Case Study on the Impact of Customer Communication on Defects in Agile Software Development. In AGILE2006, USA (2006)
23. Lowery, M. and Evans, M.: Scaling Product Ownership, Agile 2007, USA (2007)
24. Mann, C. and Maurer, F.: A Case Study on the Impact of Scrum on Overtime and Customer Satisfaction. In ADC IEEE Computer Society, USA, PP.70–79 (2005)
25. Martin, A. et al.: The XP customer role in practice: Three studies. In: ADC'04. pp.42–54, IEEE Computer Society, Washington (2004)
26. Misra, S. C. et al.: Identifying some important success factors in adopting agile software development practices. *J. Syst. Softw.* 82, 11, pp.1869—1890 (Nov. 2009).
27. Nerur, S. et al.: Challenges of migrating to agile methodologies. *Com.ACM*, pp.72–78 (2005)
28. Pikkarainen, M. et al.: The impact of agile practices on communication in software development. *J. Empirical Softw. Engg.*, pp. 303–337 (2008)
29. Schwaber, K., Beedle, M.: Agile Software Development with Scrum. Prentice-Hall (2001)
30. Strauss, A., Corbin, J.: Basics of Qualitative Research. Sage, Newbury Park, CA (1990)
31. Whitworth, E., Biddle, R.: The Social Nature of Agile Teams. In Agile'07, pp.26–36, IEEE Computer Society, USA (2007)