

# Don't Mention the 'A' Word: *Agile Undercover*

Rashina Hoda, James Noble, Stuart Marshall  
Victoria University of Wellington  
Wellington  
New Zealand  
{rashina, kjx, stuart.marshall}@ecs.vuw.ac.nz

**Abstract**—Agile methodologies openly value customer collaboration. However, customers may not openly value the term Agile methodologies. Consequently some Agile practitioners have resorted to Agile Undercover, a process of being Agile internally while keeping the customer unaware of the project's Agile nature. This process creates tension between the theoretical basis of the Agile method and the need to carefully evolve customer expectations of their own commitment. We explore Agile Undercover in the context of the Indian software industry, and discuss both why some practitioners don't mention the 'A' word, and how these practitioners attempt to stay credibly Agile in such scenarios.

**Keywords**—Agile methods, Agile Undercover, Grounded Theory

## I. INTRODUCTION

One of the defining features of Agile development is the customer's involvement with the project. Customer collaboration and rapid feedback are vitally important in Agile projects [1]. However, we have discovered that real-life Agile practitioners (from development/vendor organizations) sometimes practice '*Agile Undercover*'. We define '*Agile undercover*' as the practice of *development companies* following Agile practices internally, while their *customer* remains unaware that Agile practices are specifically being followed.

This paper explores why some Agile practitioners practice *Agile Undercover*. Reasons identified include dealing with customer skepticism, transitioning to Agile, managing large customers, negotiating contracts, and dealing with lack of synchronization.

We also investigate the repercussions of leaving the customer out of the developmental process, and describe how these Agile practitioners overcome those repercussions. Our analysis is based on Grounded Theory research conducted in India with eight different Agile practitioners from seven different organizations.

In section II we describe our research methodology, the role of the customer, and a description of our participants. The rest of the paper presents the results of this analysis. In section III we discuss why practitioners may decide to keep their customers unaware of internal Agile practices, and section IV discusses the practitioners' techniques for

handling *Agile undercover*. Section V outlines related work, and the paper concludes in section VI.

## II. RESEARCH METHOD

We will first recap the nature of the customer's role in Agile development, and then described the major characteristics of the Grounded Theory qualitative research method. Finally, we will describe the projects and practitioners participating in our research, and briefly describe our means of capturing and analyzing the data.

### A. The Agile Customer

The customer's involvement in the traditional waterfall method of software development was typically limited to providing the system/product requirements in the beginning, and feedback towards the later delivery stages [9, 15]. As a consequence, there was limited interaction between the customer and the development team [10].

Agile methodologies aimed to enhance and expand the customer role within the entire development process [5]. The customer is expected to be involved by writing user stories, discussing product features, prioritizing the feature lists, writing test cases, and providing rapid feedback to the development team on the features delivered in each of the iterations [11]. Ideally, the customer organization would spare one resource (at a time) and designate them as the on-site customer representative. This customer representative is ideally an individual who has both thorough understanding of the project requirements as well as the authority to take strategic decisions [11].

### B. Grounded Theory

Grounded Theory is a qualitative research method developed by Glaser and Strauss in 1967 [2]. Glaser and Strauss define the method as:

*"A grounded theory is one that is inductively derived from the study of the phenomenon it represents. That is, it is discovered, developed, and provisionally verified through systematic data collection and analysis of data pertaining to that phenomenon."*

Grounded Theory allows us to study the social aspects of human behavior and as such is well suited to our research on software development teams using Agile methodologies which focus on people and interactions [1]. Grounded

Theory is increasingly used to study the social nature of Agile teams. Cockburn [4] used Grounded Theory to research the relationship between people and methodology as a part of his doctoral thesis. More recently, Martin, Biddle and Noble [5], Ferreira, Noble and Biddle [6] used Grounded Theory to explore social aspects of Agile teams such the role of the XP on-site customer and the role of interaction designers in Agile projects.

Grounded Theory differs from other research methods in that it does not test a hypothesis but rather allows the theory to emerge from the data [3]. There are two main versions of Grounded Theory, such as Glaserian and Straussian. The main point of difference between the two versions is that Glaser believes that the research problem is uncovered as the collected data is analyzed, whereas Strauss and Corbin advocate setting the research question in advance [7]. We use the Glaserian version of Grounded Theory [8] and as such started out with a general area of interest – Agile project management – rather than begin with a predefined research problem. Our aim is to generate new theory rather than confirm existing theory.

### C. Participants and Projects

We interviewed eight practitioners from seven different software organizations in India. The organizations used combinations of Scrum and XP, and the interviewed participants were mostly Scrum Masters and Agile coaches. There were several Agile teams within these organizations. The teams used Agile practices such as iterative and incremental development, daily standup meetings, frequent delivery of software, continuous integration, testing and more. In order to respect the participants' confidentiality we will refer to the participants in this paper by numbers only.

The organizations were involved in the development of web-based applications, front and back-office applications, and also offered software development and Agile training services. The average project duration was three to six months and the average team size was ten people.

### D. Data and Analysis

Our approach was to ask the participants open-ended questions projects in semi-structured, face-to-face interviews regarding the challenges and strategies of managing Agile projects. The interviews were audio recorded where the participants permitted, otherwise hand-written notes were taken. We also observed the workplaces and several Agile practices such as daily stand-up and release planning meeting.

We analyzed the data using Grounded Theory's constant comparison method. This method involved going through the transcripts of the interviews line-by-line. We recorded a short sentence – called a 'code' – for each topic or idea being discussed in the transcript, and then we compared and contrasted the codes with each other to extract common patterns or 'concepts'. Further rigorous analysis and comparison of these concepts led to 'categories', a higher level of abstraction of the data. Practicing *Agile Undercover* emerged as an important category from our analysis, alongside other categories such as negotiating contracts [13]

and dealing with team issues. The final theory generated through our research will explain the relationships between these major categories.

The rest of this paper presents the results of our data collection and analysis.

## III. WHY GO UNDERCOVER

We discovered that Agile practitioners practice *Agile Undercover* for a variety of reasons: dealing with skepticism; transitioning to Agile; dealing with large customers; negotiating contracts; and coping with a lack of synchronization. We will now go into more detail on each of these reasons. Section IV will then discuss how practitioners practice *Agile Undercover*.

### A. Dealing with Skepticism

The enthusiasm amongst software developers for Agile practices is – in some quarters – matched by skepticism on the part of customers. Even as practitioners claim that Agile practices represent a significant paradigm shift for the software industry in recent times, customers often wave it off like just another new buzzword.

*“On customer side, every five years they see new trend...[customers] don't care, [they] are seeing so many things.”* - Practitioner P2

Some of the Agile principles are not as readily understood by customers as originally intended, and invite customer concerns. One example is the 'fail fast' [12] concept of Agile methods, which suggests that if a project is not viable and is certain to fail, then using Agile will make this fact apparent a lot faster. The intended benefits of this – that minimum time and effort may be wasted on such a project – are not always apparent to the customer, who then becomes prejudiced about Agile.

*“Agile terms...say option to fail early - believe me customers don't want to hear it. Customers do not want to admit that there could be some problem with the nice idea they put on paper. Forget about fail early, we don't want to fail at all!”* - Practitioner P2

Another example of this skepticism can be seen when customers were unwilling to openly commit to the amount of collaboration time that the team requested. One of the practitioners shared that in cases where the customer was not willing to spare extra time to be involved with the project, the team would ask the customer to either provide one representative from their organization or wait until the team's queries were clarified by the customer.

*“customers [show] reluctance to participate in sprint meetings...got no time. [Make a] deal – [either] make one person available or wait.”* - Practitioner P6

Not all customers are enthusiastic about setting aside extra time to be involved in the development of their

product. This sentiment is conveyed in the following comments.

*“That involvement from the customer side...that kind of involvement, they do not want. Because they think okay we give you project now you do it; if we have to be so much involved we’d do it ourselves!”* - Practitioner P2

*“What is point of meeting? [there is] resistance to change [and] involvement.”* - Practitioner P6

Customers hesitated in taking time out for participating in meetings, discussions, and feature demonstrations. Because they were unaware of the benefits of being closely involved in the development of their products, they perceived it as an overhead that they had to incur specially for Agile methodologies. Thus, practitioners had to struggle to convince customers of the benefits of their involvement. Failing this, they opted to follow *Agile undercover* with such customers.

#### B. Transitioning to Agile

Some software development companies who were internally moving from traditional to Agile methodologies used *Agile Undercover* during their transition phase. This time of transition can be a significant challenge, as roles and rules both undergo changes to adapt to Agile. For some practitioners it is a time of chaos and confusion, and one coping strategy was to first successfully adopt and transition into Agile at their team levels before inviting the customer to get involved in the Agile practices. An analogy of this situation would be cleaning up and getting the home tidy before inviting guests over.

During internal adoption of Agile, practitioners still have their normal project commitments to take care of alongside understanding and accepting the changes that come along with Agile adoption. One of our practitioners who was in the process of transitioning had the following comment to share.

*“...fundamental framework which has to be put in place, that’s when it starts hampering everybody...They [team] not only have this process to initiate - this Agile transition thing - they still have this never changing project commitments.”* - Practitioner P5

Since the company did not want to lose business with existing customers, they chose to go undercover and adopt Agile internally as a first step. They wanted to get their Agile act together before publicly declaring themselves as an Agile company. During this phase of Agile adoption and transitioning, practitioner P5’s team worked on projects internally following Agile practices without the customers’ knowledge.

*“we are looking inwards right now...we don’t want our (customers) to give so much time, they are better off focusing on what the product should be rather than keeping tabs on what the engineering force is doing.”* - Practitioner P5

The Agile practices grew from the inside out, starting from the team level and gradually involving the customer. They chose to implement Agile practices internally and put a robust fundamental Agile framework in place before they decided to go public with their Agile status.

#### C. Dealing with Large-Sized Customers

Practitioner P3 provided an example of another reason for using *Agile Undercover*. The Agile company was a small organization (less than 100 employees) and their customers were much larger enterprises. In this situation, the small Agile company was unable to assert their own Agile identity and practices on the much larger customer organizations. So, in an effort to sustain business with the large customers and at the same time continue to follow Agile practices, the Agile companies decided to keep the customers unaware of the internal Agile adoption and practice.

*“...in none of the [three] cases the customer was aware of Agile, they didn’t really want to do Agile, and 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. From the very high level waterfall projects...but what we had done was - the company had taken charge of the projects - we had made it Agile - internally following Agile.”* - Practitioner P3

Here the practitioner disclosed that the organization that they worked for had successfully completed three projects where the customer was unaware of the company’s internal Agile processes. The large customer organization was not bothered about the internal operations of the software company and enforced their traditional style of working at the higher level. However, the software company was able to follow some Agile practices internally on the ground level, without the knowledge of the customer.

*“Initially you begin with customers with small to medium domain so it appeals very well to them because they are the ones who are not very clear about their requirements and big enterprises tend to like waterfall better than ...because they believe more contracts rather than agility.”* - Practitioner P3

Practitioner P3 found it easier to practice Agile with smaller organizations since their requirements were more susceptible to change, whereas larger organizations had more firm requirements and ways of working which made it all the more difficult to introduce Agile to them. The software company chose to follow *Agile Undercover* with larger organizations.

#### D. Negotiating Contracts

The customer’s expectation of fixed price/scope/time contracts also influences some Agile practitioners’ decision to use *Agile Undercover*.

*“With Agile it’s difficult to do fixed price projects. Agile talks about embracing change, can’t do fixed price projects with changes coming in.”* - Practitioner P3

*“All they [customers] have done is fixed price for last 20 years...very difficult to say it will not be fixed price.”* - Practitioner P3

Agile practitioners try to convince customers to think beyond fixed bid contracts by sharing with them the different advantages of using Agile methods. These advantages include the customers’ ability to introduce change in requirements, to be intimately involved in the development of their product, to incrementally receive working software on a regular basis, and to benefit from greater visibility via the transparency that Agile methods offer.

However, the customers are not always flexible about their contractual agreements and as a result Agile practitioners face serious repercussions with respect to loss of business.

*“...no match between what Agile says and the way they [customers] wanted. Yes, we lost business.”* - Practitioner P3

*“...fixed price - trying to get rid of it - doesn't work well with Agile.”* - Practitioner P1

There is a tension between customers’ habitual preference for classical fixed price/scope/time contracts, and Agile teams’ desire to derive the most benefit from their Agile methods by having the freedom to keep the constraints of price, scope, and time flexible [13]. However, in real life projects, customers greatly value the perceived stability and predictability as far as the project duration, cost, and deliverables are concerned.

It was difficult for our practitioners to change the mindset of the customers regarding fixed bid contracts. So, often they ended up following Agile internally on a team level while the legal contract governing the project remained traditional fixed bid. Thus, the customer was kept unaware of internal Agile practices and it appeared like a traditional project to them.

#### *E. Dealing with Lack of Synchronization*

Teams also sometimes followed *Agile Undercover* when there were organizational barriers to synchronizing the activities of the team and the customer.

Practitioner P3 discussed an interesting experience where their software organization was following Agile practices and was delivering new functionality to the customer every two weeks. The customer in this case was responsible for deploying the functionality onto the final end-user platform. This customer was not used to Agile and while they appreciated rapid iterative development and delivery of their products, they did not deploy those features on to the final platform in the same rapid iterative manner. As a result, the live environment that the customer controlled was always lagging behind the staging environment where the Agile team used to develop and deploy working software.

*“Customer was not used to releasing the software as soon as it was given to the... that is the biggest problem we faced with them. You can take a set of requirements and do those things iteratively but how do you make the customer take the piece of software [and] deploy it every 2 weeks? They were not used to doing that...it used to remain on shelf...the development work always behind the staging by 2 to 3 iterations.”* - Practitioner P3

This is an example of how the customer’s processes caused what was perceived by the Agile team to be unnecessary delay and frustration.

Not all Agile companies are able to positively influence the customers to change their work culture to synchronize with that of Agile teams. Often practitioners are unable to successfully sell Agile and may be forced to remain content with following Agile on an internal team level without expecting the customer to be truly involved in the project in the manner the Agile recommends.

#### IV. AGILE UNDERCOVER TECHNIQUES

We have discussed reasons why some Agile teams try to do *Agile Undercover*. We now discuss some of the techniques they employ to achieve *Agile Undercover*. If a software development company claims to be following Agile, they cannot afford to ignore customer involvement and feedback entirely. So how do development companies practice *Agile Undercover* while maintaining their Agile values?

We discovered that our participating Agile practitioners have devised some interesting strategies to overcome the issue of improving customer involvement and feedback while keeping them unaware of internal Agile practices. The two main strategies are scheduling demo sessions and using a customer proxy. Not all of these techniques would work in the context of all of the original reasons for going *Agile Undercover*. For instance demo sessions may not work well if the original reason for going *Agile Undercover* is due to synchronization problems. However our practitioners report that the strategies do work for their particular situations.

##### *A. The Demo Session*

One of the ways practitioners secured rapid feedback from customers while using *Agile Undercover* was by making frequent releases of the developed software to the customer and requesting feedback. At the same time the practitioners focused on the growth of their internal Agile framework at the team level.

*“We were making frequent releases to customers and asking for feedback so customer was not aware okay... being more and more Agile with that.”* - Practitioner P3

After delivering the software, representatives from the customer organization were invited to demo sessions every fortnight. While the customers were not used to reviewing working software on a regular basis, they appreciated this concept. The customers were unaware that delivering working software on a frequent basis and receiving customer

feedback were normal practices of most Agile methods. Impressing customers with frequent progress - without explicitly identifying that as an Agile feature - may help overcome any initial skepticism on the customer's part derived from any association with a new method.

*"They liked what we were doing because they were not used to some additional features very fifteen days. We gave demo after fifteen days. We were getting four-five people from client organization in the demo. They were pretty impressed with that concept...they were pretty happy with the results."* - Practitioner P3

By frequently delivering software and requesting feedback, the team was able to secure rapid feedback from customers while practicing *Agile Undercover*.

### B. Using Customer Proxy

Another way to practice *Agile undercover* and still manage the required customer involvement was through the use of a customer proxy. Agile teams nominated a member of their own team to act as the customer proxy (or Product Owner in case of Scrum). The customer proxy was responsible for providing and prioritizing the product requirements on the basis of their interactions with the real customers. They served as a link between the customers and the development team. The customer proxy would communicate the customers' requirements to the team and clarify the team's questions from the customers.

*"...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."* - Practitioner P8

The use of customer proxy may not be easy to adopt. As practitioner P1 stated, it requires time for people to adjust the idea of playing proxy and interacting directly with the developers.

*"Customer proxy never talked to developers felt something wrong happening. Initially wanted PMs [Project Managers] on call...[now] finding it easier."* - Practitioner P1

A customer proxy may help in situations where the team is going through an Agile transition; is involved with large-scale customers; or is working with customers who are unwilling to make large time commitments.

## V. RELATED WORK

The practice of using customer proxy is not unique to *Agile Undercover*. It is also employed in situations where the customer is unable (as compared to unwilling) to provide a representative from their own organization due to time or budget constraints [14]. Grisham et. al discuss that customer organizations feel the burden of customer involvement when

they are required to spare an employee for Agile projects [15].

In the case of Scrum, a member of the development team may play the role of the Product Owner, prioritizing the Product Backlog (list of features to be implemented) in consultation with the real customers [16]. They also participate in the Scrum Review (demo and review of the product developed each iteration) and sometimes in the Scrum Retrospective (process review).

Sometimes a proxy may work to support a Product Owner. Judy et. al describe the use of a proxy to support the Product Owner [14]. The proxy was a member of the team and an experienced Scrum Master. The use of a proxy allowed the Product Owner to fulfill their role with the minimum of time commitment and allowed the team to benefit from the continuous presence and involvement of the Product Owner proxy.

Another situation where the Product Owner role may be derived from the development team is when the 'customer' is in fact the end-user. Lowery et. al report on experiences in scaling Scrum at the BBC [17]. The 'customer' in this case was the end-user of the internet services provided by BBC's online iPlayer project. As such the role of the Product Owner was delegated to member from within the different development teams. This was akin to the use of a proxy customer that we noticed in our participants' teams.

Similarly, the use of traditional fixed bid contracts at the senior management level in conjunction with Agile practices at the team level is also common practice. Several practitioners have recommended optional or flexible scope contracts to be a possible solution to the problem of using fixed bid contracts for Agile projects. Beck's optional scope contract makes quality a constant while keeping the scope negotiable [18]. It allows the customer to prioritize features and introduce change.

Sutherland suggests the use of a 'change for free' clause in traditional fixed price contracts, which allows the customer to exchange or swap features so long as the total work remains the same [19].

Deemer et. al also recommend that in order for traditional contracts to work for Agile projects, there must be a built-in buffer for change [20]. The different solutions proposed above are evidence of the fact that negotiating contracts for Agile projects is a real problem that practitioners face as we discovered in our research.

## VI. CONCLUSION

We have conducted Grounded Theory qualitative research in the context of the Indian software industry. The data collected during the research identified a new trend amongst Agile practitioners to use *Agile Undercover* because of the practical difficulties of managing customer expectations and the customers' own processes. The major factors that lead to the practice of *Agile Undercover* are: difficulty in eliciting adequate customer involvement and enthusiasm; dealing with lack of synchronization; dealing with large customers; and negotiating contracts. In some cases, the practitioners used *Agile Undercover* as a reaction

to the lack of customer involvement, while in other cases they utilized this practice on a temporary basis until they successfully understood and adopted Agile at their own organization before involving the customer.

We need to conduct further study with a larger set of participants to determine how widespread *Agile Undercover* is amongst other Indian Agile practitioners and projects, and this paper and research method cannot make generalized claims about the degree of *Agile Undercover* that typically occurs across the industry. Similarly, it would be interesting to explore whether these same problems and strategies exist in other countries, and what role culture plays in this matter.

Is *Agile Undercover* a viable strategy to handle the problems identified in this paper? Our practitioners have stated their satisfaction with their own take on *Agile Undercover*, however future work needs to explore whether these solutions are optimal and/or sustainable. Future work should also explore the long-term implications on the team/customer relationship brought on by the strategies.

#### ACKNOWLEDGMENT

We would like to thank all those Agile practitioners who have participated in our research. We sincerely hope that our results prove beneficial to your understanding and practice of Agile methodologies.

#### REFERENCES

- [1] M. Fowler, J. Highsmith, The Agile Manifesto, *Software Development* Issue on Agile Methodologies, <http://www.sdmagazine.com>, 2001.
- [2] A. Strauss, B. Glaser, The Discovery of Grounded Theory, Adline, Chicago, 1967.
- [3] A. Strauss, J. Corbin, Basics of qualitative research: grounded theory procedures and techniques, Sage Publications, California, 1990.
- [4] A. Cockburn, People and Methodologies in Software Development, University of Oslo, Norway, 2003.
- [5] A. Martin, R. Biddle, J. Noble, "The XP Customer Role in Practice: Three Studies", Proc. Agile Development Conference (ADC 04), IEEE Computer Society, Washington DC, 2004, pp. 42–54.
- [6] J. Ferreira, "Interaction Design and Agile Development: A Real-World Perspective", Victoria University of Wellington, New Zealand, 2007.
- [7] Gerry Coleman and Rory O'Connor. Using grounded theory to understand software process improvement: A study of Irish software product companies. *Inf. Softw. Technol.*, 49(6):654–667, 2007.
- [8] J. Smith, *Qualitative Psychology: A Practical Guide to Research Methods*, SAGE, 2008.
- [9] G.K. Hanssen, and T.E.Fægri, "Agile Customer Engagement: A longitudinal qualitative case study". *ISESE '06: Proc. ACM/IEEE international symposium on Empirical software engineering*, ACM, Rio de Janeiro, Brazil, 2006, pp. 164-173.
- [10] X. Wang, Z. Wu, and M. Zhao, "The Relationship between Developers and Customers in Agile Methodology". *Computer Science and Information Technology, 2008. ICCSIT '08, Reliability Eng. Center, Guizhou Univ., Guiyang, 2008*, pp. 566- 572.
- [11] S. Fraser et al., "The role of the customer in software development: the XP customer - fad or fashion?". *OOPSLA '04: Companion to the 19th annual ACM SIGPLAN conference on Object-oriented programming systems, languages, and applications*, ACM, New York, USA, 2004, pp. 148- 150.
- [12] K. Beck, *Extreme Programming Explained: Embrace Change*. Addison-Wesley Longman Publishing Co., Inc. 2000.
- [13] R. Hoda, J. Noble, and S. Marshall, "Negotiating contracts for Agile Projects: A Practical Perspective". *Proc. Agile processes and eXtreme Programming in Software engineering (XP2009)*, Sardinia, Italy, 2009, pp 186-191.
- [14] K.H. Judy and I. Krumins-Beens, "Great Scrums Need Great Product Owners: Unbounded Collaboration and Collective Product Ownership". *Proc. Hawaii International Conference on System Sciences*, Waikoloa, Hawaii, 2008, pp. 462-462.
- [15] P.S. Grisham, and D.E. Perry, 2005. Customer relationships and Extreme Programming. *Proc. Workshop on Human and Social Factors of Software Engineering (HSSE 05)*. ACM, New York, NY, 1-6. doi: <http://doi.acm.org/10.1145/1083106.1083113>
- [16] K. Schwaber, M. Cohn, and E. Derby, "Scrum Ceremonies". Scrum Alliance website, doi: [http://www.scrumalliance.org/pages/scrum\\_ceremonies](http://www.scrumalliance.org/pages/scrum_ceremonies).
- [17] M. Lowery and M. Evans, "Scaling Product Ownership", *Proc. Agile 2007*, Washington DC, 2007.
- [18] K. Beck, D. Cleal, "Optional Scope Contracts", 1999, doi: <http://www.xprogramming.com/ftp/Optional+scope+contracts.pdf>.
- [19] J. Sutherland, "Agile Contracts: Money for Nothing and Your Change for Free", 2008, doi: <http://jeffsutherland.com/scrum/2008/10/agile-contracts-money-for-nothingand.html>.
- [20] P. Deemer, G. Benefield. "The Scrum Primer. An Introduction to Agile Project Management with Scrum", 2007, doi: [http://scrumtraininginstitute.com/home/stream\\_download/scrumprimer](http://scrumtraininginstitute.com/home/stream_download/scrumprimer).