The Offshoring Elephant in the Room: Turnover

Darja Šmite, Blekinge Institute of Technology

Rini van Solingen, Delft University of Technology

Panagiota Chatzipetrou, Örebro University / Blekinge Institute of Technology

Abstract: Staffing software projects with engineers from best-cost locations has become a commonality. However, distributed development remains practically challenging with many recurring problems, such as decreased productivity, low quality, and high unforeseen extra costs. One main underlying reason for these challenges is high employee turnover, although often overlooked. In developing locations such as India turnover is significantly large due to personal benefits from 'job-hopping'. Why is turnover such a problem? Should companies stop sourcing to countries with high turnover or are there known remedies? This research puts turnover of software engineers in India in the spotlight and derives strategies to address it. We share experiences from two industrial cases, discuss important variables for portraying the actual turnover state and its negative impacts. Furthermore, we put forward ten recommendations for actively reducing turnover itself and lowering its negative consequences.

Although offshoring projects are challenging, they promise the benefits of competitiveness, getting access to large labor pools, and lowering development costs. As a result, globalization of software companies and their partnerships continues to grow[1]. The success rate of distributed offshoring projects, however, remains low and the realization of expected benefits is often debated[2-7]. Empirical studies reported indicate that quality and performance problems are usually caused by the underestimated complexity of the work, gaps in competences, long time for acquiring the necessary knowledge in the newly established sites and high employee turnover[5-7]. Turnover and its consequences in offshoring collaborations is the focus of our study. Based on two recent industrial cases of offshoring to India, we empirically demonstrate that turnover, especially retention of engineers in the first two years of employment, is a real challenge and share practical

recommendations to address it.

Turnover in the Indian software industry

India has earned a distinct identity as a software since the early years of millennium[8]. *Location India* attracted numerous contracts from customers worldwide, both large[9] and small[4]. However, Asian job market is recognized for the high employee turnover rates[9,10] due to abundant job opportunities[2,3] (see Sidebar 1 for definitions). The average yearly turnover in India ranges between 20-40%[1,9-10].

Turnover is not only an India-specific challenge; it is one of the top risks in global software development in general[1]. While some studies report Europe to have lower turnover rates than developing nations[1], market monitoring reports suggest that turnover is also a burning issue for European countries. The UK experiences 20% turnover and the average length of service of 4,7 years for engineering jobs[15]. Evidently economic growth could be a more dominant driver for turnover than the geographic location.

SIDEBAR 1: Measures to Capture Staff Changes

Motivated by the challenges with portraying the true picture of staff turnover in our research, we outline the concepts important for having good control over personnel changes based on the human resource management literature[13].

Companies keep track of employees within a fiscal or calendar year. *Attrition* and *turnover* are core concepts often used interchangeably but are not the same. When *attrition* occurs, vacancies remain unfilled, while *turnover* refers to replacement of lost employees. Employee turnover can be *external* when employees leave the company, or *internal* when employees leave the current job and move within the company. Turnover can happen *voluntarily* when employees leave willingly, or *involuntary* when the company terminates employment. The *turnover rate* for a period of time is calculated by dividing the number of the employees who left, by the average number of employees during that period.

Why is Turnover a Problem?

Some may wonder why turnover is such a problem. Fresh hires cost less and "new blood" brings new ideas and a fresh perspective[3]. Yet, research recognizes the disrupting effects of high turnover for organizational performance. Software companies shall be especially concerned because knowledge-intensive industry significantly depends on the intellectual capital[5]. So, replacing software engineers often has economic, operational and behavioral implications.

Economic impact
A loss of employees results in a necessity to hire, train and onboard new people as they climb up the learning curve[5]. This carries direct costs:
- cost of recruitment – advertising, interviewing candidates or contracting recruiting agencies;
- knowledge transfer and training cost – training of the new hires onshore or offshore;

- travel cost – sending onshore trainers offshore, and bringing offshore trainees onsite, and additional exchange visits to increase the familiarity among the personnel;
- cost of mentoring and support – support during the learning process onshore or offshore.

Operational impact

Turnover problems are linked with the inability to achieve acceptable productivity and quality levels[2,4-6]. A study of turnover impacts in large engineering companies in the UK found that the average lost productivity to replace an engineer resulted in a substantial cost of over $6,500[15]. When employees leave, the productivity suffers not only because they don't contribute anymore but also because those who stay spend time on onboarding the new hires; this effort is often underestimated[3,6]. Productivity drops even if replacements are found fast. Thus, turnover always affects the schedule commitments and the value the team is set to deliver.

Behavioral impact

A poor motivation for introducing offshoring significantly impacts the behaviors of onshore and offshore employees[11]. When onshore developers have fears about their employment, they become self-protective and purposefully make it difficult for offshore colleagues to contribute. Poor collaboration makes offshore developers concerned and less motivated to engage. Continuous turnover also prevents building trust and often leads to frustrations on both sides[11].

Turnover in DutchCo and SwedCo

Our findings from studying two cases, DutchCo and SwedCo (names are anonymized) offshoring to India, suggest that turnover is a burning issue (see turnover data in Table 1).

Table 1. Case profiles and staff changes.

		DutchCo	SwedCo
Profile	Company location	The Netherlands	Sweden
	Company size	Small	Large
	Industry domain	Telecom	Telecom
	Indian partner	InVend (external)	SwedCo (internal)
	Time frame	2009-2016	2014-2017
Staffing	Employed (start)	4 (Jul 2009)	11 (Aug 2014)
	Employed (end)	17 (Dec 2016)	72 (Jun 2017)
	Average no of employees	Approx. 8	Approx. 50
	Employees left	23	64
Turnover	Total turnover rate	295% (in 8 years)	128% (in 4 years)
	Yearly average turnover rate	36%	35%
	Total external turnover rate	205%	62%
	Total voluntary turnover rate	257%	Unknown
	Total involuntary turnover rate	38%	Unknown
	Total internal turnover rate	90%	66%
Reasons	External: Personal reasons	26%	Unknown
	External: Better opportunities	26%	Unknown
	External: Termination	13%	Unknown
	External: Unknown	4%	43%
	Internal transitions	30%	57%

DutchCo is a small Dutch company founded in the mid-90s as a software product and service supplier for customers in the telecom domain. DutchCo outsourced to an Indian vendor, InVend, since 2009. The relationship started by employing one in-house and one offshore team in the first five years and evolved to four mixed teams in the last two years. We found that employee turnover was a root cause of many problems. During eight years of collaboration, they experienced 36% average yearly turnover and 288% in total. To retain 17 developers at the end of the eighth year, the company hired 40 and lost 23.

SwedCo, a large international company headquartered in Sweden develops a wide range of software-intensive products in the telecom domain. We studied turnover in one product that originated in the 90s. Due to growing demands of capacity and necessity to implement customer-specific features, developers from several SwedCo sites worldwide were involved. In 2014, SwedCo onboarded developers from their Indian site. The Swedish developers were set to gradually move to other assignments, while the Indian site was set to become the main development force. Three years later Swedish developers still supported the Indian site who struggled with performance. The average yearly turnover in India was 35%, and the total turnover in the third year reached 128%. To have 72 developers in 2017, the company hired 136 developers and lost 64. In our study, we found that the initial perception of profitability, employee motivation, attitude, productivity and job satisfaction in both collaborations over time decreased (See Figure 1). Although project success is not only affected by turnover, we observed that it played a significant negative role.

Economic impact

In DutchCo, training costs were not explicitly accounted because of small scale of collaboration with employees being trained more or less informally. Travel costs, however, were accounted. In the first five years, these reached 5% of the total salary-based yearly

costs of offshore employees and doubled in the following years. The support effort (feedback from testing) was estimated in the first five years of collaboration as 4% of the total salary-based yearly costs of offshore employees.

In SwedCo, the continuous mentoring and support through code reviews and consultation were significant and resulted in an extra cost of 36% of the total salary-based yearly cost of offshore employees during the first year of collaboration, and 24% in the second year. These needs were motivated by large amounts of complex legacy code.

Operational impact

In DutchCo, performance problems emerged both onshore and offshore, because of large amounts of technical debt. As one offshore developer explained: *"The code we started with was completely messed up. It was garbage. If you touched one part of the code, you had ten things breaking"*. Naturally, it took a long time for new developers to learn. This is why turnover led to a long period of unsatisfactory performance and delayed profitability when evaluated at the end of the fifth year[5]. Offshore developers were perceived to progress slower on a learning curve (three years) than their peers in the Netherlands (two years). One reason being the inability to retain offshore developers long enough to accumulate the critical mass to locally train the new hires. In five years, the average amount of unproductive time among the InVend developers constituted 52%, decreasing over time from 95% in the first year to 25% in the fifth year. But although DutchCo and InVend addressed turnover, time does lead to people leaving. By the end of the eight-year period, team changes happened again, resulting in performance problems and decreased trust. Though the actions taken during the years, including reengineering the product, made it easier and faster to learn, turnover still significantly affected performance.

In SwedCo, the productivity of the new site was around 25% of the mature developers[6] and remained almost unchanged for three years primarily because of high turnover and overall product complexity. As the onshore manager explained: *"At the end of the third year, tasks get done, but the amount of customer value delivered is limited"*. Although not many developers left SwedCo (external turnover), developers who were good were promoted or moved (internal turnover), and therefore did not directly contribute to the code any longer. Also, onshore mentors had to spend most of their time supporting the new hires, and the loss of their productivity was noticeable.

Behavioral impact

In DutchCo, offshore employee turnover significantly impacted the satisfaction with InVend, and the morale of offshore developers. The productivity problems injured trust in the supplier capabilities and affected the relationship. Offshore developers who stayed were motivated to do their best to satisfy the customer and repair trust, even if that meant working extra hours. As one offshore developer explained: *"We made 16-18 hours a day 5-6 days a week. And that for almost a 6-month period. That did deliver results, a real partnership, and satisfaction at work. But it was not sustainable!"* The situation improved only after reengineering the product and shortening the learning curves (in the seventh year). The relationship improved and a trustworthy collaboration was established for a while. However, when coupled work in mixed teams suffered from the distance and turnover in the eighth year, things changed again. The behavior of the new hires injured trust, onshore developers became frustrated, and the consequent escalation to InVend management had a direct negative impact on job satisfaction and morale offshore.

In SwedCo, we found turnover to have a significant impact on the supporting roles. In the beginning, onshore developers were motivated to engage in the knowledge transfer, because they were to move to more challenging assignments. However, after three years of not seeing significant productivity increases due to turnover, some onshore mentors became so frustrated with the necessity to continuously re-train the replacements, that they left SwedCo. Ironically, turnover triggered turnover. As the onshore manager explained: *"Developers want to work with other developers who are skilled. If being forced to work a lot with less skilled offshore developers, they become unmotivated".* Turnover also put more stress on the retained employees offshore, lowering satisfaction.
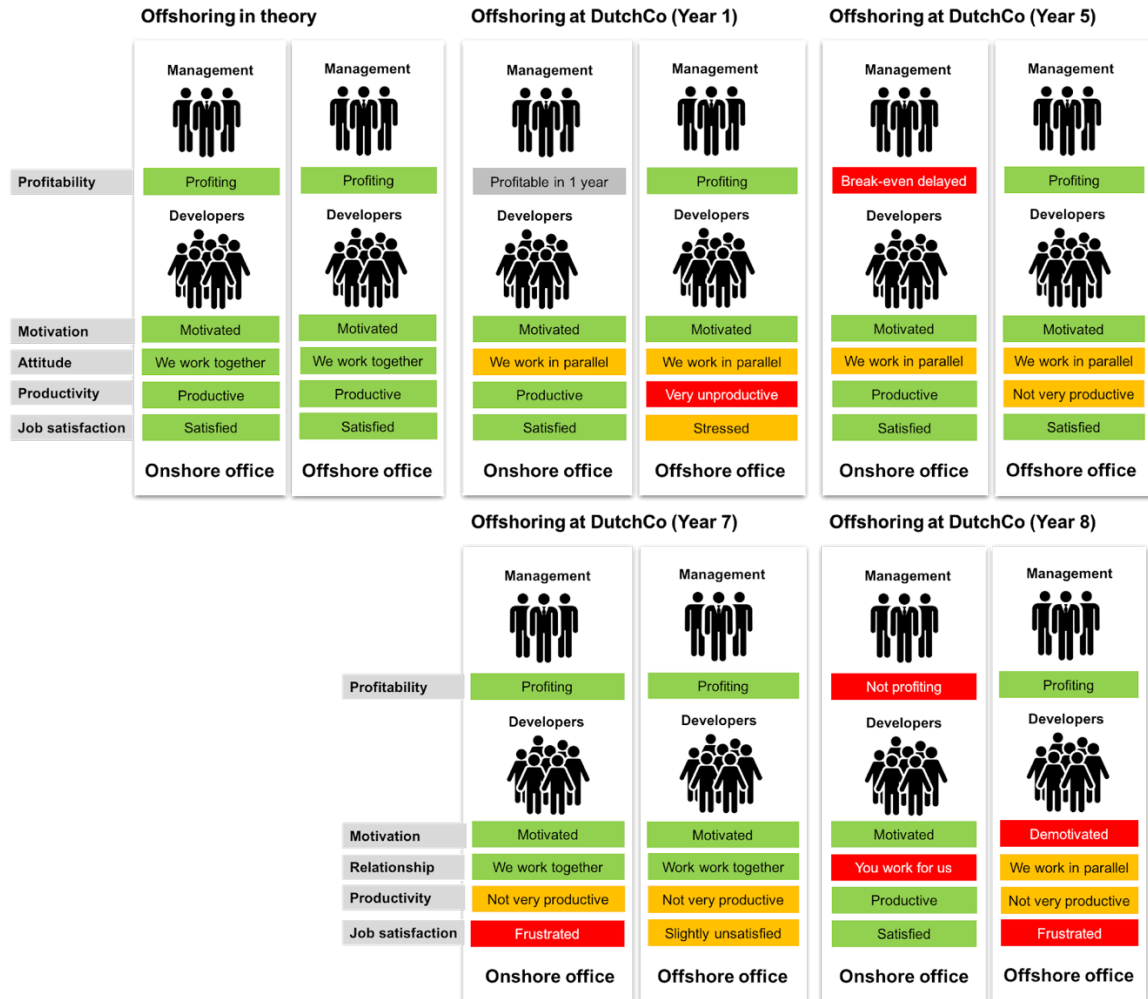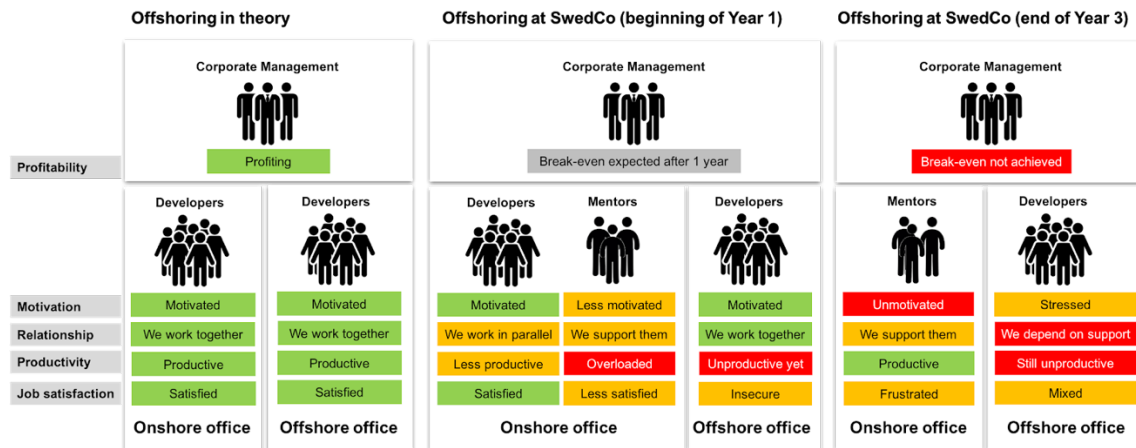
Figure 1. The impact of turnover visualized for DutchCo and InVend.

Strategies to address the turnover

To address the turnover problems, companies may choose to invest in practices reducing the actual turnover or reducing its negative consequences.

Reducing the turnover

Experience from 93 offshore projects from a leading Indian vendor shows that retaining people is a challenge[10].

Recommendation #1: Provide additional extrinsic motivation. To combat the retention issue in India, companies use various extrinsic motivators on top of traditional means of competitive compensation and pay, including the high quality of work and work life, perks, loans, stock options, overseas assignments, and career progression[8]. Career potential has been recognized as a crucial strategy[1,5,14].

Recommendation #2: Ensure learning and growth opportunities. Career potential is not only about promotions to managerial roles but also about satisfaction of intrinsic needs, e.g., becoming a recognized expert. Important intrinsic motivators also include continuous learning opportunities and increasing employability[8], satisfying the needs for independence and variety[14].

Recommendation #3: Provide intellectual challenges and avoid intellectual frustrations. While routine work and a lack of intellectual challenge increase turnover[3, 14], too high thresholds for becoming a valuable contributor in the project may also significantly increase the turnover[12]. Being a mediocre developer for years is a likely cause of frustration for any developer, and potentially the main reason why many leave.

Recommendation #4: Foster sense of belonging and relatedness. Outsourcing contracts rarely focus on promoting relatedness and a sense of belonging, which are important motivators[14]. Yet, to succeed, companies will increase integration between the sites, enable direct personal relations between the onshore and offshore engineers and treat

the offshoring counterparts as allies and partners rather than contractors or resources.

Recommendation #5: Screen out 'job hoppers'. DutchCo and SwedCo data show that many engineers leave within the first two years. The pattern of changing companies is known as 'job-hopping', and can often be identified during the hiring process already. Turnover can be prevented directly by actively screening out 'job hoppers' during the hiring process. To do so, managers will include a peer-to-peer interview with other engineers to discuss personal drive and ambition, or invite a candidate to a social event with a team, during which personal career plans can be discussed informally.
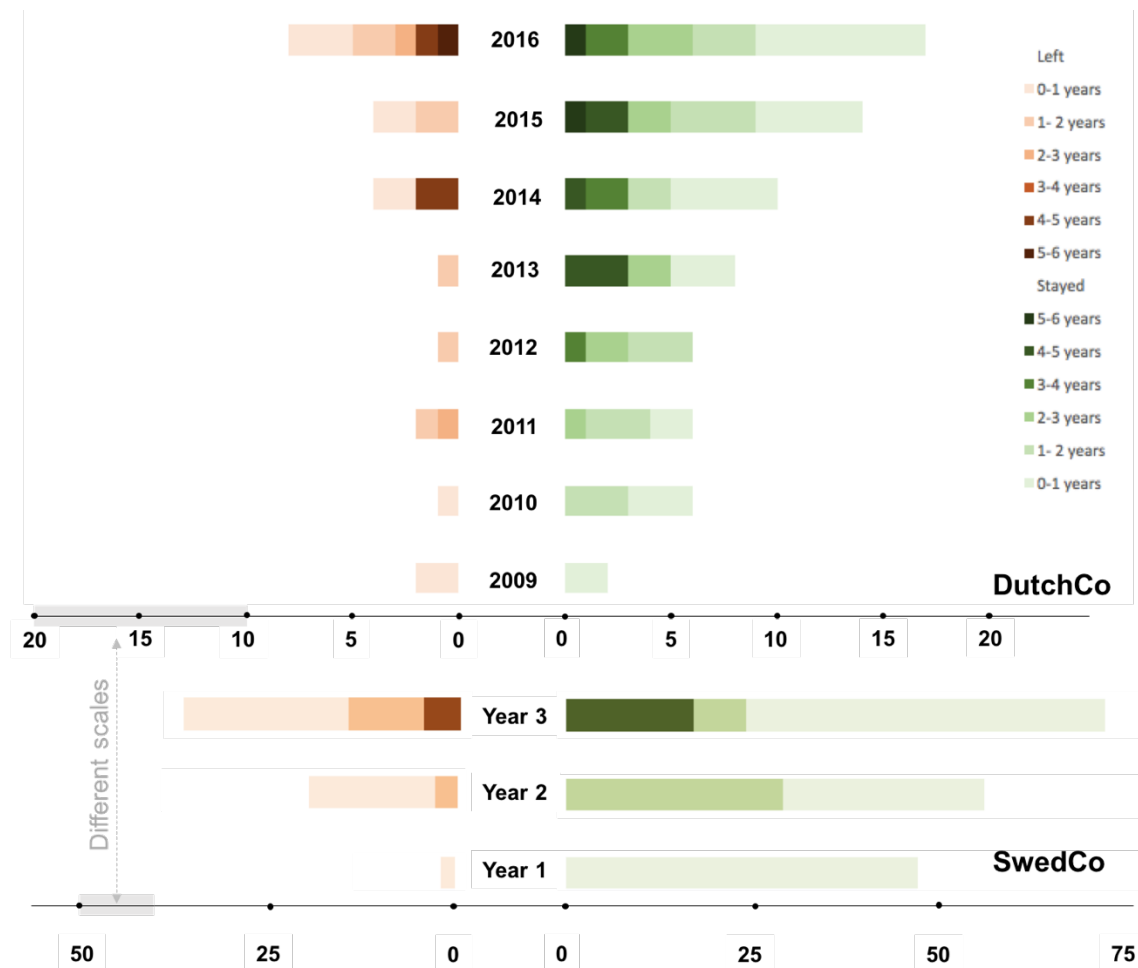
Figure 2. Employees staying and leaving by competence levels.

DutchCo and **SwedCo** implemented many strategies to increase motivation and reduce retention – included value-added functions, reduced product complexity, implemented modern technologies and ways of working, integrated the sites and evolved their relationship with InVend into a partnership. Also InVend helped to address turnover by screening out job hoppers. Despite these efforts, the career opportunities remained limited, so, after a certain period, people still left. In the end, it seems that turnover could not be eliminated from the DutchCo setting. SwedCo had more control over their employees in

India than in an outsourcing setting such as with InVend. SwedCo could influence the future of employees in the company and provided many challenges, responsibility and good career, growth and travel opportunities. Yet, the work appeared to be too challenging, and the long learning curves led to significant frustration and increased employee turnover.

Recommendation #6: Monitor turnover. A more general lesson learned from these cases is that isolated turnover rates may be misleading. DutchCo and SwedCo perceived the same rates at different times differently, without understanding why. Our results suggest that although turnover rates are important, *who* quits matters most. Involuntary turnover, when engineers who are not good enough are fired, will have lesser operational impact. On the other hand, the loss of experienced engineers has serious implications. We, therefore, recommend monitoring employee experience structures and situations when the critical mass of experience deteriorates (see Figure 2). To do so, companies should maintain detailed turnover records, including the numbers of and reasons behind external/internal turnover, voluntary/involuntary turnover, and monitor the actual staffing experience structure.

Reduce the impact of turnover

If reducing the turnover rate remains a challenge, we suggest making it a smaller problem. As a management representatives from DutchCo said: *"Attrition is a given fact. We can influence it to lower it, but we cannot exclude it totally. Anticipating attrition is a much better strategy".*

Recommendation #7: Increase the volumes. Small offshore assignments tend to be not only more prone to turnover, but also under-prioritized by the offshore vendors[4]. Scaling up the number of offshore developers helps to create a stronger critical mass of people with experience.

Recommendation #8: Use employee buffers. To reduce the impact of turnover companies may plan employee buffers[1], i.e. proactively onboarding 'shadow resources' as potential future replacements. This strategy is common in experienced vendors. Furthermore, prior research from offshoring to Eastern Europe, India, and China shows that offshoring to mature vendors yields better results because process maturity serves as a safety net to assure the right training and good management practices[1].

Recommendation #9: Support onboarding of the new hires. To speed up the process of learning and acquiring the necessary product knowledge, companies will provide sufficient support and mentoring[6]. This includes dedicated expert time, exchange visits and investments in the training material. In some cases, refactoring and restructuring software code may significantly alleviate the onboarding.

Recommendation #10: Find suitable tasks. A cheaper strategy is to offshore tasks that do not demand large amounts of training and high onboarding efforts. So tasks based on well-defined processes and clear deliverables are more likely to succeed than longitudinal complex projects[7]. The rule of thumb here is that the average duration of stay of an employee shall be longer than the time it takes to become productive.

SwedCo and especially **DutchCo** have had a number of strategies in place to reduce the impact of turnover. DutchCo reduced the scope of what developers needed to learn, reengineered the product, simplified the tasks (epics), as a part of the new agile ways of working, increased the number of exchange visits (from two to eight per year) and scaled up the number of offshore developers to create a stronger critical mass of people with experience. All these strategies combined, together with InVend employee buffers, helped developers become productive within one year and offshoring benefits surface earlier. **SwedCo** experience does not span as many years to have many opportunities for experimentation. SwedCo followed one particular strategy to lower the impact of turnover: active involvement of onshore experts in supporting the new hires. As long as the onshore support remained, the impact of turnover was said to be under control, even though this strategy was expensive.

Conclusions

Organizations heavily invest capital and energy to increase productivity, time-to-market, customer satisfaction, and overall performance. Offshoring is often a strategy to achieve this, be it for access to talent and knowledge, lower wages, or proximity to customers. However, a crucial, turnover, is often ignored. We, thus, depict turnover as the "elephant in the room" in software engineering offshoring.

Taking a look at our cases, DutchCo needed five years to fully comprehend the need to address high turnover, while SwedCo tried to combat external turnover by promoting people within the company, ending up with significant project-level losses due to internal turnover. We suspect that this is either because the companies largely underestimated the negative consequences of turnover, or accepted it implicitly as a fact of life. At the same time, the damage caused by the turnover became a significant cost factor. Furthermore, its magnitude only became visible after a period of years.

To help companies fight turnover, we created a list of ten recommendations to decrease turnover rates and decrease the negative impacts of turnover, as a deliverable of this research. Our findings can help others in making informed decisions regarding offshoring, taking turnover impacts explicitly into account. It can also help making practical changes in daily operations to reduce the negative impacts of turnover.

Our research is country-specific and, similarly to the majority of research focusses on the offshoring destination India[1,2,3,5,6,8,9]. Is India special? Maybe, but we can't be sure yet. Also, there are signs that turnover has become a challenge in the western world too[15]. Especially economic growth seems to be a more dominant driver for turnover than the geographic location itself. Other important variables might be present too. Some studies, for example, indicate that outsourcing relationships experience higher staff turnover than captive sourcing[5,14]. We, therefore, recommend studying other geographies and paying attention to organizational and project characteristics.

SIDEBAR 2: Research approach

Empirical cases from this article have been published in two separate reports, on DutchCo[4] and on SwedCo[5], focusing on the true costs of offshoring. Here, we focus on turnover. We added data from three additional years in DutchCo and one year in SwedCo. We reevaluated the DutchCo collaboration with InVend and captured the evolution at SwedCo through additional semi-structured interviews specifically focusing on understanding the role of turnover. In DutchCo, we interviewed the CTO and one of the offshore developers. In SwedCo, we interviewed an onshore manager on several occasions.

The study is empirical and thus has limitations. Our findings by no means are generalizable to an isolated company offshoring to India. Product complexity in our cases determined the operational impact, which for simple products or tasks might significantly differ. Finally, many additional factors, including work distribution strategies, cultural differences, temporal dispersion, local leadership approaches and contractual agreements, could potentially influence the behavioral impacts of turnover.

References

1. Ebert, C., Murthy, B. K., & Jha, N. N. (2008). Managing risks in global software engineering: principles and practices. In *IEEE International Conference on Global Software Engineering*. pp. 131-140.

2. Conchúir, E. Ó., Ågerfalk, P. J., Olsson, H. H., & Fitzgerald, B. (2009). Global software development: where are the benefits? *Communications of the ACM*, *52*(8),127-131.

3. Dibbern, J., Winkler, J., & Heinzl, A. (2008). Explaining variations in client extra costs between software projects offshored to India. *MIS quarterly*, *32*(2),333-366.

4. Moe, N. B., Šmite, D., Hanssen, G. K., & Barney, H. (2014). From offshore outsourcing to insourcing and partnerships: four failed outsourcing attempts. *Empirical Software Engineering*, *19*(5),1225-1258.

5. Šmite, D., & van Solingen, R. (2016). What's the True Hourly Cost of Offshoring?. *IEEE SW*, *33*(5),60-70.

6. Šmite, D., Britto, R., & Van Solingen, R. (2017). Calculating the extra costs and the bottom-line hourly cost of offshoring. *Proceedings of the International Conference on Global Software Engineering,* pp.96-105.

7. Šmite, D., Calefato, F., & Wohlin, C. (2015). Cost savings in global software engineering: Where's the evidence. *IEEE Software*, *32*(4),26-32.

8. Moitra, D. (2001). India's Software Industry. *IEEE Software*, *18*(1),77-80.

9. Bagchi, S. (1999). India's Software Industry: The People Dimension. *IEEE Software*, *16*(3),62-65.

10. Gopal, A., Sivaramakrishnan, K., Krishnan, M. S., & Mukhopadhyay, T. (2003). Contracts in Offshore Software Development: An Empirical Analysis. *Management Science*, *49*(12), 1671-1683.

11. Piri, A., Niinimäki, T., & Lassenius, C. (2012). Fear and Distrust in Global Software Engineering Projects. *Journal of Software: Evolution and Process*, *24*(2),185-205.

12. Ford, D., & Parnin, C. (2015). Exploring causes of frustration for software developers. In *Proceedings of the International Workshop on Cooperative and Human Aspects of Software Engineering*, pp.115-116.

13. Blau, G., & Boal, K. (1989). Using job involvement and organizational commitment interactively to predict turnover. Journal of management, 15(1),115-127.

14. Bass, J., Beecham S., Razzak MA., & Noll J. (2018). Employee Retention and Turnover in Global Software Development: Comparing In-house Offshoring and Offshore Outsourcing. Proceedings of the International Conference on Global Software Engineering.

15. Pontin, G. & Pearson, G. (2015). The Retention Gap – What it is and how to tackle it. *Association for Consultancy and Engineering and EngTechNow*. Whitepaper.

DARJA SMITE is a full professor of software engineering in the Blekinge Institute of Technology, where she leads research efforts and education on global software development. She has led a number of nationally funded research projects related to the effects of offshoring for Swedish software industry, with such partners as Ericsson, Spotify, ABB, DXC, Emerson Process Management, and Boss Media. Her other research interests include large-scale agile software development, and software process improvement. Smite received a PhD in computer science from the University of Latvia. Contact her at darja.smite@bth.se.

RINI VAN SOLINGEN is a part-time full professor of global software engineering at the Delft University of Technology, where he also heads research and education about worldwide distributed software teams. In addition, he's the chief technology officer of Prowareness We-On (www.prowareness.nl). Van Solingen received a PhD in management science from the Eindhoven University of Technology, and an M.Sc. in Computer Science from Delft University of Technology. He's a member of IEEE. Contact him at rini@rinivansolingen.nl or d.m.vansolingen@tudelft.nl.

PANAGIOTA CHATZIPETROU is an Assistant Professor at the department of Informatics at Örebro University School of Business in Örebro, Sweden, where she belongs to the *Centre for empirical research on information systems (CERIS)*. She is also part of the *Software Research Engineering Lab (SERL)* at Blekinge Institute of Technology. Her research interests include applications of statistical methods to quality problems in software engineering and especially to requirements engineering, the exploitation of human factors that ultimately determine the quality of a software product and the product development, decision support systems, large-scale agile (and global) software development, and behavioral software engineering. She received her Ph.D. in Informatics from the Department of Informatics, Aristotle University of Thessaloniki (AUTh), Greece. Contact her at pchatzip@gmail.com.