

# Examining the Impact of Adopting Inner Source Software Practices

**Noel Carroll**

Lero, National University of  
Ireland Galway, Galway,  
Ireland  
noel.carroll@nuigalway.ie

**Lorraine Morgan**

Lero, National University of  
Ireland Galway, Galway,  
Ireland  
lorraine.morgan@nuigalway.ie

**Kieran Conboy**

Lero, National University of  
Ireland Galway, Galway,  
Ireland  
kieraen.conboy@nuigalway.ie

## ABSTRACT

Open Source software (OSS) has been highly prevalent in both practice and research. Given the value and effectiveness of OSS development to date, practitioners are keen to replicate these practices inside their respective corporations. This application of OSS practices inside the confines of a corporate entity has been coined Inner Source Software (ISS). While ISS presents many benefits, little is known about the opposing tensions that arise as a result of transitioning from a closed to an open software development environment. Such environments are increasingly under pressure to embrace more open and collaborative principles internally, while simultaneously managing operations in a tight and controlled manner. As part of this study, we conducted 20 interviews with international ISS experts across 15 global organisations. We uncover 13 core tensions that arise from the adoption of open principles in closed software practices. Based on these emerging results, we present new insights on the implementation of strategies to overcome competing tensions from openness in software development. We present some recommendations, which also call for fundamentally new research directions.

Author Keywords

Inner Source; Open Source; Tensions; Software Practice.

## ACM Classification Keywords

Software and its engineering (Software creation and management; Collaboration in software development).

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from [Permissions@acm.org](mailto:Permissions@acm.org).

*OpenSym '18*, August 22–24, 2018, Paris, France

© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM.

ISBN 978-1-4503-5936-8/18/08...\$15.00  
<https://doi.org/10.1145/3233391.3233530>

## INTRODUCTION

Organisations continuously strive to improve their software development processes in order to decrease costs, reduce development time, improve quality and remain competitive. Over the past decade there has been an increase in the adoption of Open Source Software (OSS) practices inside an organisation, something termed ‘Inner Source Software’ (ISS). Similar to OSS development, ISS applies an open, acquiescent model of collaboration and communication, and has been referred to as a good example of intra-organisational open innovation [1]. Achieving these benefits however, presents many challenges for traditional software development practices. Indeed, changing how software teams operate without truly understanding how practice is impacted upon can lead to unforeseen organisational tensions and ultimately prevents organisations from continuously improving software practices. This warrants the need to investigate tensions associated with what may be viewed as the paradox of adopting OSS principles in closed software environments.

### Traditional Software Development

There have been significant research efforts to examine the key differences and possible incompatibilities between different software development paradigms. Several emerging phenomena reflect attempts to address these problems such as, agile software development, DevOps, Scrum, XP, Kanban, continuous development. While these methodologies have proved extremely effective for software development, one of the constant challenges is the need to overcome teams working in silos and the barriers that prevent sharing of knowledge and code [2]. This has given rise to the need for more open, collaborative and transparent practices in software practices such as ISS.

The contribution of this research is that we identify the powerful aspects of tensions between closed and open practices that play a dominating role in successful adoption. Such insights have not been explored within a software engineering context and sheds new light on how ISS practices can be better managed.

### From Open To Inner Source

OSS may be described as original source code that is made freely available, redistributed and modified [3]. Given its success to date, organisations are increasingly adopting OSS development practices to support their internal

software development processes, something coined ISS [4]. ISS development practices must be tailored within the confines of a specific organisational context. However, ISS is not a defined methodology per se, but rather, a philosophy, using OSS principles that are considered to add value to an organisation's software development practices [5]. Existing research indicates that ISS can enjoy improved engineering and management practices to enable different business units to improve their performance, achieve better results and gain higher levels of team performance [6] [7]. However, there is limited research that has examined what tensions surface as a result of implementing ISS. This raises important questions regarding the management of ISS initiatives and the need for new management models to identify and manage tensions within ISS environments. Hence, our research questions for this study are as follows:

- RQ1: What are the core tensions that arise in the adoption of ISS practice?
- RQ2: How do tensions impact on the management of ISS practices?

This research aims to support management in their ability to realign strategies to optimise the success and value realisation of ISS initiatives.

### **TENSIONS IN OPEN SOFTWARE PRACTICE**

While the concept of tensions is highly complex and has been interpreted in many different ways, it remains something that is not well understood in the software engineering domain. For example, Huxham and Beech [8] refer to tensions as being “concerned with the choice between alternative forms of management practice”. Tensions can exist in many forms. For instance, Smith and Lewis [9] categorise tensions as paradoxes, dilemmas or dialectics. Tensions within the organisation play a crucial role “in determining the path that an organisation's development will follow. In turn, each new structural arrangement realigns the power structure and creates new tensions” [10]. Indeed we can identify the emergence of such tensions from the seemingly paradox of principles in modern software development practices. For example, both OSS and ISS environments have become a prominent strategic opportunities for software organisations [11]. Moreover, OSS and ISS environments allow organisations to alter their path dependence by transforming their strategic intent and innovation capabilities [1], [12]. However, such a transformational shift can also generate new yet undocumented tensions. Understanding the nature of tensions is crucial for the planning of new software practices which forms the basis of this empirical study.

#### **The Nature of Tensions**

Tensions can be either disruptive where for example they can lead to some form of breakdown, or they can be beneficial, by fostering competition and challenges with management involved in the process of continually resolving tensions [13]. Within a management context, for example, Huxham and Beech [8] revisits the notion of

‘good management practice’ in terms of tensions between apparently contradictory pieces of good practice advice. They present an interesting conclusion in that due to often conflicting theoretical evidence “the good practice prescriptions are not easily implementable in practice...[and]...the membership of collaborations tends to be in a constant state of flux, with individuals, organisations and their priorities rarely remaining stable [8; p.70-71]. However, in the case of ISS, we frequently learn about ‘practices’, while other descriptions of this phenomenon describe ISS ‘principles’. As such, the paradox is not resolved, but it often mismanaged. Importantly, tensions are always present and managing such paradox requires explicit exploration, rather than suppression of various tensions that arise between the polar opposites resulting from emerging software practices.

Within an OSS context, the concepts of principles serve as the foundation for a system of belief or behaviour, which from a practical perspective can make it difficult to implement and indeed monitor software environments. For example, many organisations experience difficulty shifting to new technologies, moving away from familiar practices or norms, changing strategic paradigm, breaking out of prevailing patterns of decision-making, adjusting their product architecture and learning from experience. The results emerging from this study present new insights on the importance of managing tensions and is considered crucial to ISS success. This also highlights the need to incorporate strategies to overcome competing tensions of openness in software development. We examine these factors further within an ISS context in order to identify a) what tensions exist; b) whether the tension is constructive or deconstructive; and c) how tensions are (mis)managed within a software context.

### **METHODOLOGY**

We conducted a reflective qualitative research process for this study. We conducted 20 semi-structured interviews with international ISS experts. We analysed the interview data using qualitative analysis techniques and identified preliminary themes that emerged from these interviews. This enabled us to construct and consider the application of emerging theoretical and conceptual developments. We summarise these findings in the next section.

Our first task was to identify key ISS stakeholders who are experts to participate in interviews. Drawing interviews into a coherent whole, gives meaning to and explains each element of our findings, i.e. thematic and narrative coding. More importantly, to examine how events unfold, narratives allow a form of sequential causation. The research participation criteria led to a combination of practitioners (ISS working experience) and academics (ISS research experience with various industry projects) were selected to provide a rich mix of participants across a range of sectors including telecommunications, software, financial, consultancy and education. As well as ensuring participants

had extensive experience in terms of number of years and were involved in many projects or roles. As the unit of analysis was at the project level, all practitioners were required to be ‘substantially’ involved in two or more ISS projects. This led us to conduct 20 semi-structured interviews that, for example, comprised of the following research questions:

- In a general sense, what does ISS mean to you?
- Describe the core tensions of ISS practices in your organisation.
- How would you describe this tension [a list of tensions described from questions 2 and the interviewer explored these individually in greater details]?
- Describe the key lessons learned regarding the tensions encountered across the ISS practice.

The interviews began with some general open-ended questions around the company’s usage or planned usage of ISS in addition to questions about the company and interviewee background and experiences. Subsequently, interviewees were asked more specific and direct questions and they felt comfortable sharing their experiences and uncovering tensions emerging from various ISS initiatives. Interviews were carried out in person or by telephone and were recorded. Interviews lasted between 40 and 80 minutes. When required, follow-up questions were asked by phone or email. Data analysis was undertaken using various coding techniques. In the initial phase, ‘open coding’ was used to determine the main ideas and themes in each transcript. This allowed us to categorise the core tensions emerging from ISS initiatives.

### **SUMMARY OF FINDINGS**

While organisations often report that they need or want to adopt OSS principles internally, our findings identify some of the key tensions that emerge in the management of various degrees of openness in closed environments. Specifically, we address the first research questions by identifying 13 key tensions associated with adopting OSS principles within an ISS environment (summarised in Table 1). We outline key insights that interviewees shared with us regarding their experience with such tensions. We also address the second research question by examining how these tensions impact on the management of software practices, i.e. moving from a traditional management approach to an innovation management approach. We summarise the key tensions in Table 1 and offer some insights on management strategies to overcome tensions across ISS practices.

### **Management Strategies to Overcome Tensions Across ISS Practice**

In response to the 13 tensions we identified, we outline four key management strategies at the initial stages to manage such tensions in ISS practices. It is important to stress that rather than trying to quash emerging tensions, managers should explore the cause and indeed nature of such tensions

with a view to foster constructive tensions and resolving deconstructive tensions. This can be accomplished by implementing four broad strategies that are summarised as follows:

- **Learning Strategy:** invest in core competencies and capabilities that facilitate a learning culture within an organisation. For example, our results indicate that targeting middle management to overcome any misunderstandings or sense that ISS threatens their ability to effectively manage staff and identify competitive strategies. In addition, identifying constructive tensions, e.g. competitive tensions across teams, should be fostered to improve quality, productivity, and performance across an ISS community.
- **Incentivisation Strategy:** implement incentives/rewards to foster a culture of collaboration, openness, transparency, and trust that acknowledges and rewards employee performance. Our findings indicate that it is important that management identify what it is that contributors want as a reward and introduce incentivisation schemes that are tailored to their organisation. As part of the incentivisation strategy, providing teams with clarity on how their efforts are contributing towards the success of an ISS community, e.g. through metrics or visualisation techniques, improves the sense of belonging to the ISS community.
- **Knowledge Sharing Strategy:** moving beyond a command and control model to develop a ‘safe environment’ for employees and management to feel comfortable in learning, asking questions, and sharing insights on failures and successes. Our findings indicate that this can reduce the sense of risk towards critique or being exposed to flaws by promoting a transparent work environment. This also complements the learning strategy by encouraging teams to describe experiences (e.g. successes and failures) and promote continuous improvement within an ISS community.
- **Software Analytics Strategy:** management need to adopt new software practice metrics to capture the socio-technical factors that support and sustain an ISS community. Our results indicate that this will be beneficial by shedding new insights on the relational infrastructure within the ISS community. This will also reveal where other possible tensions exist or beginning to emerge across the community. This would provide value to managers as existing metrics are considered limited and to be unreliable in reflecting the true nature of an ISS community.

Focus	Source of Tension	Description
OSS Principles within a Closed Software Environment	Inner Source Philosophy	Leads to uncertainty surrounding the value of ISS practice. This uncertainty creates tensions emerging from ISS principles promoting openness and collaboration, yet managers attempt to micro-manage teams in order to reduce risks associated with implementing software practice change.
	Code Ownership	There is a traditional sense of code ownership amongst software developers. Our results indicate that not all developers are ‘open’ or comfortable with sharing code within an ISS model depending on their expertise levels.
	Commitment to Projects	Recognition is a key factor to sustain ISS teams. However, tensions from receiving recognition for various contributions (i.e. star status) can emerge due to growing commitments across various additional software projects. In practice, this can create a bottleneck in terms of quality reviews and contributions whereby commitment leads to increased responsibility to manage ISS teams.
	Risk of Change	Depending on the organisational culture, middle managers are often tasked with coordinating software teams and are accustomed to traditional management practices. Change in software practices brings about an element of risk. Change also creates tensions amongst managers to weigh-up the ‘risk vs. rewards’ of implementing ISS.
	Inner Source Practice	ISS is based on specific practices yet in essence, the literature and evidence largely focuses on ISS principles. The terms ‘principles’ and ‘practices’ are used interchangeably and generates tensions when organisations attempt to operationalise ISS principles.
	Concept of Business Value	From a business perspective, middle managers need to ensure their team performs at a desired level. However, often due to the lack evidence on business value with ISS, tensions also emerge whereby managers remain unconvinced to take a risk in adopting a change in software practices, i.e. ISS.
	Evidence of Performance	Tensions emerge through the notion of ‘evolution’ (closely linked to tensions associated with change). The results indicate that it is important to incorporate measures/metrics that demonstrate performance improvements through ISS. This would also off-load responsibility on middle management decision-making tasks and their expectation on software team performance.
Management of ISS: from a traditional approach to an open management approach	Transparency of Practice	Transparency is a key factor in collaboration to ensure the benefit of sharing code is visible across the organisation. However, there are growing tensions between the need for transparency as it gives employees a sense of exposure within a software team. It is worth noting that such exposure can be either positive or negative – depending on the developers’ competencies, level of commitment to projects, and their desire to become ‘stars’ across an ISS community.
	Trust in Committers	Tensions emerge between team members demonstrating their software engineering expertise. For example, when code features have been submitted but little is learned on the development process, there is a need to accept the notion of ‘trusted committership’ within the community.
	Motivation of Team	A balance between the motivation to contribute code and the expectation of job performance can create tensions. This is particularly true around the quality of software development and the unpredictability of a team’s overall performance. Tensions emerge from the need to motivate teams to volunteer time towards projects that may be considered to be of less priority to middle managers.

**Table 1: Summary of Key Findings**

Focus	Source of Tension	Description
Management of ISS (continued)	Scaling Practice	Planning to scale software development in ISS environments creates tensions in choosing between the risks of scaling the software practice or choosing the ‘safer’ option to contain risk and micro-manage software teams.
	Need for Incentivisation	Tensions exist where organisations do not wish to invest in ISS using monetary incentives, yet rewards are a key necessity in ISS. As managers are familiar with controlling and regulating practices within a closed environment, they do not often offer incentives that encourage transparency or commitment to projects. However, this is largely dependent on the nature and culture of the organisation and their attitude towards rewarding employees.
	Accountability and Responsibility for Productivity	There is considerable pressure in being responsible for the success of software in various products and the success in adopting ISS practices. Central to this is the uncertainty associated with whether ISS will increase productivity and team performance that escalates tensions between accountability and responsibility to ensure productivity.

**Table 1: Summary of Key Findings (continued)**

ISS Management Strategy	Sources of Tensions
<b>Learning Strategy:</b> this focuses on addressing learning tensions and explores initiatives to build on competencies and capabilities.	<ul style="list-style-type: none"> <li>• Safety of traditional practice vs. risk of change</li> <li>• ISS principle vs. ISS practices</li> <li>• Personal motivation vs. growing expectations of management</li> <li>• Invest in learning vs. invest in incentivisation schemes</li> <li>• Scale software practice vs. management containment</li> </ul>
<b>Incentivisation Strategy:</b> this focuses on addressing belonging tensions and identifies factors that inspires and motivates software contributions.	<ul style="list-style-type: none"> <li>• Balancing reliance of ISS champions vs. overloading them with tasks</li> <li>• Contributor recognition in ISS vs. contributor responsibility in ISS</li> <li>• Openness philosophy vs. rigid management practices</li> </ul>
<b>Knowledge Sharing Strategy:</b> this focuses on organising tensions and promotes evidence-based practice and decision-making initiatives.	<ul style="list-style-type: none"> <li>• Sense of code ownership vs. increased collaborative requirements</li> <li>• Practice transparency vs. accountability for decision-making</li> <li>• Fear of the unknown vs. trust in team credibility</li> </ul>
<b>Software Analytics Strategy:</b> this places emphasis on addressing performing tensions by examining the socio-technical factors in a collaborative software environment.	<ul style="list-style-type: none"> <li>• Concept of business value vs. concept of ISS success</li> <li>• Pressure to measure performance vs. need to evolve practice</li> </ul>

**Table 2: Strategies to Overcome Competing Tensions of Openness in Closed Software Development Environments**

The implications of these findings highlight both the constructive and deconstructive role that competing tensions play in hampering and driving ISS practices. Table 2 aligns our proposed strategies with the 13 tensions we identified in the findings. We outline how the strategies complement the key tensions namely, performing, organising, belonging, and learning identified in the literature. We provide specific insights on the 13 core tensions that surfaced from our findings and we offer four broad management strategies which can address these tensions: learning, incentivisation, knowledge sharing and software analytics. In addition, these findings provide new insights in developing a research roadmap to address the critical shortcomings across ISS literature and practice.

Our research suggests that tensions cannot be completely eliminated from an organisation since tensions are fundamental to sustaining two opposing forces that support software teams and their development practices. To manage and balance such tensions, strategies may be employed to better manage them and ensure that tensions align with organisational goals, i.e. by reducing deconstructive tensions and regarding constructive tensions. Such strategies, however, must be tailored for various organisational cultures and maturity levels in adopting or scaling ISS practices.

#### **SUMMARY OF CONTRIBUTION**

The core contribution of this research is the identification of undocumented tensions associated with adopting OSS principles inside the confines of an organisation (i.e. ISS). The software engineering research and practitioner community can build on this research by developing and demonstrating mechanisms to overcome deconstructive tensions while incentivising constructive tensions.

We present 13 tensions from our initial findings and explain how this research can provide a platform to build on across the software community. This research also supports both the research and practitioner community in identifying and managing tensions in modern software organisations that are exploring the adoption or sustainability of ISS practices.

Theoretically, we will further expand on our research related to tensions in software engineering with the need to support our understanding of the intersection between people, processes and technology within a software context. This will offer us rich insights on how interactions, flows and activity generate competing tensions and how practitioners can become better informed through suitable ISS strategies from a socio-technical perspective.

#### **DISCUSSION**

Software engineering continues to be challenging for large corporations, often due to complex organisational structures, diverse processes, and scarce resources. ISS provides an alternative software practice philosophy that leverages OSS practices within the confines of an organisation. ISS is considered to be a valuable asset to

sustain competitive advantage. However, examining why competing tensions exist and how they can be overcome plays a key role in the successful adoption and scaling of ISS. While much of the literature on ISS is still in its infancy, we are mindful that in order to promote such software practices, we need to be able to translate the value in adopting ISS. We encourage ISS communities to adopt four core ISS management strategies to foster an open, collaborative and successful software practice community.

#### **Limitations of this Research**

This study is constrained by some limitations. Our data gathering approach focused on tensions categorised in literature, namely learning, belonging, organising, and performing. There may be additional categories of tensions that would enrich future research avenues. Our research participants were largely large multi-national organisations that may not allow us to generalise the results. Future research will also focus on a systematic literature review in order to identify the variety of research methods employed for ISS studies.

#### **CONCLUSION**

In this paper, we identified the core tensions of openness within a closed software environment. We describe the impacts of such tensions on managing a software environment and highlight the importance for management in becoming aware of these tensions. We explain that while the use of ISS is rapidly gaining popularity, exploiting its purported benefits is poorly understood, particularly in terms of performance and business value. Central to performance are the tensions we describe in this paper that have constructive and/or deconstructive influences on the overall performance and decisions around adoption or scaling of ISS practices.

As part of our future research, we plan to expand on, implement and test these strategies to overcome competing tensions of openness in closed software development environments. We will also adopt a suitable socio-technical theoretical lens to describe the emergence and management of ISS tensions. In this paper, we focus on an area that has been undocumented across software engineering strategies and explore how tensions compete with each other as a result of transitioning to an open software development environment.

As part of future research avenues, we call for the need to examine ways to model and manage tensions across ISS practices in an effort to develop tension resolution strategies and actionable analytics to determine the business value of ISS initiatives.

#### **ACKNOWLEDGMENTS**

This work was supported with the financial support of the Science Foundation Ireland grant 13/RC/2094 and co-funded under the European Regional Development Fund through the Southern & Eastern Regional Operational

## REFERENCES

1. Lorraine Morgan, Joe Feller, and Pat Finnegan, 2011. Exploring inner source as a form of intra-organisational open innovation. European Conference on Information Systems, Helsinki.
2. Darja Šmite, Nils Brede Moe, Aivars Šāblis, and Claes Wohlin, 2017. Software teams and their knowledge networks in large-scale software development." *Information and Software Technology* 86: 71-86.
3. Joe Feller and Brian Fitzgerald, 2002. Understanding open source software development (pp. 143-159). London: Addison-Wesley.
4. Tim O'Reilly, 2000. Open Source and OpenGL - O'Reilly Media. Retrieved June 22, 2018 from [http://archive.oreilly.com/pub/a/oreilly/ask\\_tim/2000/openssl\\_1200.html](http://archive.oreilly.com/pub/a/oreilly/ask_tim/2000/openssl_1200.html)
5. Noel Carroll, 2017. How Is Value Created Within An Inner Source Environment? In Proceedings of the 13th International Symposium on Open Collaboration (p. 22). ACM, Galway.
6. Klaas-Jan Stol, Muhammad Ali Babar, Paris Avgeriou, and Brian Fitzgerald, 2011. A comparative study of challenges in integrating Open Source Software and Inner Source Software. *Information and Software Technology* 53, no. 12: 1319-1336.
7. Øyvind Hauge, Claudia Ayala, and Reidar Conradi, 2010. Adoption of open source software in software-intensive organizations—A systematic literature review. *Information and Software Technology* 52, no. 11 (2010): 1133-1154.
8. Chris Huxham and Nic Beech, 2003. Contrary prescriptions: Recognizing good practice tensions in management. *Organization Studies* 24, no. 1: 69-93.
9. Wendy K Smith and Marianne W. Lewis, 2011. Toward a theory of paradox: A dynamic equilibrium model of organizing. *Academy of management Review* 36, no. 2: 381-403.
10. Brian Hobbs, Monique Aubry, and Denis Thuillier, 2008. The project management office as an organisational innovation. *International Journal of Project Management* 26, no. 5: 547-555.
11. Brian Fitzgerald, 2006. The transformation of open source software. *MIS Quarterly*: 587-598.
12. Joel West and Scott Gallagher, 2006. Challenges of open innovation: the paradox of firm investment in open-source software. *R&D Management* 36, no. 3: 319-331.
13. Orla O'Dwyer, Roger Sweetman, and Kieran Conboy, 2014. Exploring control tensions in IS project portfolio management, 35th International Conference on