Agile Methods as a Risk Management Strategy Tool–A FinTech Case Study

Completed Research

Anna Zaitsev
The University of Sydney
anna.zaitsev@sydney.edu.au

Abstract

FinTech companies are subjected to same regulations than traditional financial organisations but due to their novelty, these organisations can experiment with unorthodox ways to strategically manage software development risk and ensure compliance with the regulations of the financial industry. Our case study presents an example of an Australian FinTech organisation, which has successfully applied Extreme Programming method as their company-wide risk management strategy tool. The Agile methods application encompasses the different layers of risk management; method, people and process. This study discusses how Agile development method practices are utilised to respond to risks and to ensure compliance. The case organisation is shown to address both the regulatory requirements as well as fulfil the bases for risk management approach and tailoring frameworks proposed in the literature. Finally, we present a theory of strategic risk management via Agile methods. This theory illustrates how method application influences other aspects of strategic risk management.

Keywords

FinTech, risk management strategy, Agile development methods, Extreme Programming

Introduction

No business is without risks, especially a business that operates in the financial industry. However, risk management strategies that such companies employ vary greatly. The context of the business determines the sources of the risks and thus should also impact the risk management strategies. Financial organisations often manage their risk with application of control framework, for example the Control Objectives for IT and related Technology (CoBit) (Bernroider and Ivanov 2011) or similar. However, risk can be managed not only with frameworks that direct *what* to do after the software has been developed but also with frameworks that guide *how* the software is to be developed. Hence, many organisations in the financial industry have adopted the Agile software development methods (Beck et al. 2001). For example, in a recent industry survey studying the application of Agile methods, fourteen per cent of the respondents were from the financial sector with only software development industry ahead (VersionOne 2017).

Building software that aligns with risk management strategies is not limited to large organisations. On the contrary, amidst large financial organisations, which require complex control mechanisms, a novel breed of companies is emerging: small, and nowadays increasingly larger FinTech or financial technology companies are disrupting the traditional banking (Gulamhuseinwala et al. 2015). Researchers are claiming that a banking revolution (Dapp et al. 2014) or at least a transformation for the banking industry (Guo and Liang 2016) is on its way. FinTech has also very rapidly become a matter of interest for investors and consumers as well. Consumers have eagerly embraced FinTech innovations such as online banking solutions or mobile payment services (e.g. PayPal, Venmo etc.)(Gulamhuseinwala et al. 2015). Investors are rushing to back companies and projects applying novel technologies, for example, block chain ledgers, for example the Ripple ledger or other ledger project led by various consortia (Guo and Liang 2016).

These new organisations face potentially a different set of risks and apply risk management strategies of their own. New technologies create new risks and a recent report by CapGemini (Kumar et al. 2016) indicates that cybersecurity is a key driver for banking software development as well as the flexibility to meet customer needs. Agile software development proponents are addressing these needs by advocating methods that should enhance the quality of the software by producing defect-free code, secure and stable service to users and faster response to customer feedback (Highsmith 2002).

Nevertheless, the software development methods of FinTech companies are largely under-researched (save few mentions of the methods e.g. Leong et al. 2017). To understand how FinTech organisations currently address the risk concerns, we decided to approach the problem with the following research question: *How are risks and quality concerns addressed in the software development strategy in FinTech organisations?*

Traditionally, Agile development methods have been seen as incompatible with regulatory compliance. For example, McHugh et al. (2012) list several reasons that inhibit Agile adoption in similarly regulated, safety-critical fields of medical device software: lack of documentation, traceability issues, regulatory compliance, lack of up-front planning, managing multiple releases. Fitzgerald et al. (2013) trace the apparent incompatibility of Agile methods and regulated environments into the statements of the Agile Manifesto (Beck et al. 2001), which place more emphasis on human interactions and less emphasis on documentation and controls. However, more recent surveys have shown that even large, complex and regulated environments can apply Agile methods successfully (VersionOne 2017). Yet, we lack examples of such implementations and evidence of the method application success.

In order to remedy this lack of examples, we present a case study of a medium-sized FinTech company that has chosen to follow a contrarian path in the financial industry. The organisations have chosen one of the more niche Agile methodologies, Extreme Programming (Beck 1999) accompanied with rigorous DevOps practices and continuous delivery (Humble and Farley 2010), as their primary tools for risk mitigation and quality assurance. Next section will discuss the regulatory climate, risk strategy literature and briefly expand on Extreme Programming and the DevOps functions.

Regulations, Risk Strategies and Agile Development

Regulatory Compliance and IS Risk Management

Controlling risks and regulatory compliance is a multilayered issue that encompasses strategy, people, processes and technology (Racz et al. 2010). The different aspects of compliance and risk control are seen as a part of how organisations approach the expectations of their stakeholders and how the organisations manage activities to fulfil the expectations while managing risks maintaining compliance (Marks 2010).

In Australia, where our case organisation is based, the Australian Prudential Regulation Authority (APRA) determines the regulatory environment of FinTech organisations that wish to operate in the area of banking in Australia. They have stated that companies engaging in banking activities apply the following non-functional requirements for their software to manage the security risks: ***user awareness, access control, IT asset life-cycle management controls, monitoring and incident management, IT security reporting and security assurance*** (APRA 2013). The report does not specify how organisations should respond to these risks; the method is left open to interpretation. APRA's requirements cover the different aspects of compliance and risk management: people, processes, technology and strategy. These areas of risk management and risk strategy have also been addressed in the information systems literature but to various degrees.

First, the people aspect of risks and compliances is related to the opportunistic behaviour of the employees or other stakeholders (Sharma 1997), which is then linked to security policy compliance (Herath and Rao 2009). The human aspects are studied to have a significant influence on security and data reliability (Bulgurcu et al. 2010; Harrison and Jan 2017). There is a specific field of behavioural information security research that investigates the human elements impacting information security (Crossler et al. 2013). Human resourcing, employing the right people, can help to reduce risks. For

example, the actions of top management and their influence on the organizational culture have been observed to have a role in employee risk compliance (Hu et al. 2012).

However, most of the studies focus on human actions creating the risks (Crossler et al. 2013). Where the role of humans is often seen as the cause of the risks, processes and technology or methods are the components that can be applied to manage strategically the human-induced risks in information systems development and operations. The suggested approaches range from simple solutions such as risk lists to more complex risk-strategy and analysis models. In the most simple risk management process, one lists risk in priority order and ranks these risks according to their severity and probability (Barki et al. 1993; Ropponen and Lyytinen 2000). Next step of the risk analysis is links the risks with specific risk management strategies (Lyytinen et al. 1998). Finally, in a more complex analysis, the risks are first listed and then aggregated to create different risk profiles. These aggregate profiles are matched up with aggregated solutions and then allocated resources according to the severity of the risk profiles (Donaldson and Siegel 2001).

Based on these different approaches and case studies, Iversen et al. (2004) proposed an approach for both risk management and to risk management tailoring, tailoring that applies specific information systems context to the risk management. First, their risk management approach suggests that organisations need to investigate four different areas when they are assessing risks of new information systems initiatives: I. Impacted areas. II. Affected processes, tools and techniques. III. The proposed actions taken and the organisation, management and conduct of the initiative. IV. Who is involved, i.e. who are the actors. When these factors are known and the risks are characterised, the teams can analyse the risks, prioritise the actions and take actions.

Suggested actions are then presented at a high level, as an aggregate of different risk-mitigation strategies. There are five potential actions organisations can take to mitigate risks: 1. Adjust the Mission, e.g. the goals of the action that introduces risk and if these goals can be adjusted to minimize risk impact. 2. Modify the Strategy, e.g. can introducing iterations or phases minimize the impact. 3. Mobilize, e.g. are the beneficiaries sufficiently committed and engaged. 4. Increase Knowledge, e.g. the members of the organisation might require education on the subject. 5. Reorganize, e.g. how are the risky information system initiatives organized, managed and conducted.

Risk management strategies have been a concern for the Agile development method proponents from the very beginning. One of the prominent early models that applied a risk focus approach was Boehm's (1988) Spiral model of software development. The Spiral model suggests several risk assessment points during the development and strongly advocates that the risk management aspect should drive the development. Agile software development methods follow similar logic, albeit when reading only the Manifesto for Agile software development (Beck et al. 2001), the risk management strategies, embedded into the methods, are less obvious. Hence, we need to take a deeper look at how Agile software development methods, Extreme programming in particular, address risk and can be used as a risk management strategy.

Extreme Programming

Extreme Programming (XP) is one of the earliest and highly influential Agile methods. The method has been popularised by Beck (1999, 2001) and it has been applied alongside the Scrum method (VersionOne 2017). One of the key practices within the XP frame is the method of pair programming. Pair programming is a method of code development where two programmers work on the same piece of code. The pair shares one computer with two screens and two keyboards. The shared space and shared work are aimed at enabling the effective distribution of knowledge via instant, face-to-face communication between the members of the organisation. Consequently, less documentation is generally required (Beck 1999).

Where developers have embraced other practices of Extreme programming without scruples, the practice of pair programming divides practitioners and academics alike, with its benefits debated within the Agile practitioner circles (Salge and Berente 2016). However, there are multiple studies that discuss the benefits of pair programming: the code has fewer defects and the code is formatted better (Cockburn and Williams 2000). However, the practice of pair programming is demanding and thus not for every team. For example, a significant portion of Agile development is nowadays conducted in globally distributed teams (VersionOne 2017) and is thus incompatible with pair programming, which requires constant collocation.

DevOps and Continuous Delivery

One of the more recent organizational changes that came with the advent of Agile methods is the creation of the DevOps teams (compound of software **dev**elopment information technology **op**erations). Traditionally, the technology operations team is the team that oversees the hardware side of the information systems whereas the developers are focused on the software development. By combining the responsibilities of these two teams, a DevOps team is trying to break down the silos between these two parties (Hüttermann 2012).

In Agile organisations, DevOps teams are often responsible for the practice of continuous delivery. Continuous delivery, sometimes called frequent delivery, is a practice of delivering new features, fixes and changes of the product to the end users at a frequent and constant pace (Humble and Farley 2010). The aim of continuous delivery is to deliver better-quality products faster to the markets with lower costs and fewer delays (Humble and Farley 2010).

Research Methods

Case background

FinTechCo is a medium-sized software development company. The company was founded 14 years ago and has pioneered novel banking and payment solutions in Australia since its inception. Unlike large banks, where development is mostly outsourced, FinTechCo develops all products in-house and the organisation consists mostly of developers and other technical experts. Other business areas include marketing, customer services and legal teams. Everyone working for FinTechCo is located in the same premises and working from home was described to be an exception, not commonplace.

The software development work undertaken by FinTechCo is organised by product offerings. The company does not split the work according to projects, practising incremental product development. The organisational hierarchy at FinTechCo was described as very flat. My informants likened the culture of FinTechCo as a successful reflection of the values and principles of the Agile Manifesto (Beck et al. 1999). The application of the Agile method, Extreme Programming (Beck 1999; 2000), was a dominant force throughout the entire existence of the organisation.

Data Collection

The study was conducted via semi-structured, open-ended interview question as suggested by Walsham (1995). The interviews were either conducted at the premises of FinTechCo or at nearby cafes. Each interview was recorded and later transcribed and interview notes were taken during the discussion (Walsham 2006). The roles of the interviewees and their contributions to the risk strategy are summarised in Table 1.

Roles	**Risk Strategy Contributions**
Company Founder, Sales Lead	Overall organizational strategy planning
Head of Engineering (two interviews)	Development method advocacy
Head of Risk Management	Strategy monitoring and alignment with emergent risks
Head of Internal Audit	Strategy alignment with regulations
Two DevOps managers	Ensuring that end-user software is accountable
Business Product Owner	Ensuring that requirements meet regulations
Six Team/Delivery Leads Two Developers/Designers	Enacting risk strategy via development methods

Table 1. Interviewees and their strategy contributions

We began our interviews in April 2016 and conducted the last interviews in September 2016. In total, we conducted 16 interviews with the management, technical leads and practitioners at FinTechCo. The goal of the interviews was to examine the Agile software development practices applied by the organisation and discuss how such practices were addressing the needs of risk management.

Data Analysis

After the interviews were transcribed, we conducted an analysis on the themes and topics discussed by our informants. By applying open coding, we broke down each discussion into themes and then performed axial coding by comparing how each informant discussed each theme and what were the emerging linkages between the topics and risk strategy and management (Strauss and Corbin 1990). The findings of the study are presented in the next section.

Findings And Discussion

The risk strategy at FinTechCo was partially dictated by the APRA regulations, partially designed to best fit the chosen Agile method: Extreme programming. The risks were managed on two different fronts. First, the quality controls of the code development were deeply entrenched in the practice of pair programming, the chosen way of working for the product development teams. Second, the DevOps team ensured that the high-quality code was deployed to the end users in the appropriate and regulations fulfilling way.

Product Development Collaboration

Product development at FinTechCo was organised to support the pair programming practice. Collocation of the team and customers, internal or external, in an open workspace with minimal physical barriers was designed to ensure that pair programming and collaboration between all stakeholders was frictionless (Beck 1999).

The pair programming practice was the primary way of distribution of the know-how and tacit knowledge between the members of the organisation. When the developers were writing their code in pairs, there were always two or more pairs of eyes on every line of code. This was said to improve the quality and maintainability of the code. The recruitment process was specially tailored to identify which personalities would fit the teams and the intensive pair programming practice. We were told that the new recruits were screened for pair programming compatibility, including the testers and designers who had to support the development teams.

The requirements the designers, developers and testers were working with were stored both on physical walls in the form of user stories written on sticky notes and in virtual form in an Agile requirements management systems called Jira. The system assigned each story a ticket number and the story was written in so-called virtual cards, which were accessible by all relevant stakeholders. From the developer perspective, documenting the development into tickets fulfilled the regulatory requirements and also aided the collaboration between the developers, the testers and the product managers, as well as other stakeholders who had access to the ticketing software.

DevOps Collaboration

The other party directly responsible for the quality and risk management at FinTechCo was the DevOps team. The role of the DevOps was to ensure that the code that was being deployed to production, for end users, was free of defects and did not compromise the integrity of the products. One of the development team leads explained the different roles:

The DevOps team engaged with the development teams with a modification of the pair programming, paring on the changes made in the production environment. The DevOps and developers were both applying automation to ensure that the product was less prone to human error. All the different ways FinTechCo managed their risks and complied with regulations via Extreme Programming method are collected in Table 2 below. There are six different aspects to risk strategy, further discussed in the next section.

Agile method	**Implementation**	**Benefits (APRA regulation)**	**Corroborating Evidence**
Extreme Programming	Hiring practices, methods application support, showcases, 'TeDTalks', stakeholder collaboration	User awareness	*'Being open minded to different ways of thinking. No ego. People that are coming with ego hopefully don't pass the interview process. And willingness to help people, I think is a big thing …'—DevOps lead*
Pair programming	Each piece of code developed by two people, i.e. always two eyes on code Testing and design activities conducted in pairs/threes	User awareness, quality	*'We will start a piece of work. We'll pick up a task off the board. You pair up and each pair has two monitors, two keyboards, two mice, but it's one computer … They just share between each other who is controlling the keyboard at which particular time.' – Developer*
Issue tracking	User stories stored in the ticketing software No issue can be deployed without a ticket	Access control, IT asset life-cycle management controls	*'We commit software via a Jira number and that Jira number has a subject on it and that would, that subject would be, might be a storyline… So, there's clear evidence of what's been built and when it was built.'—Sales manager*
DevOps collaboration	System changes conducted in pairs User stories stored in the ticketing software DevOps tools and scripts monitoring system health	Access control, IT asset life-cycle management controls, monitoring and incident management	*So, we don't pair program. We pair with changes … Any change in production requires it to be paired. So, it's two people from the operations who implement a change. So, either they will sit together and discuss and think about the actual problem that they're trying to solve … and then implementation…'—DevOps lead*
Controlled deployment	System changes conducted in pairs No issue can be deployed without a ticket	Access control, IT asset life-cycle management controls	*'We have an operations team that puts, works solely in production. Developers don't get access to production because they want to separate those concerns. We write the codes and we release it, we build it, and then the DevOps puts it into production.'—Team Lead*
Continuous delivery	Changes applied as they come to the pipeline DevOps tools and scripts monitoring system health	IT asset life-cycle management controls, monitoring, IT security reporting and security assurance	*'There are two benefits of automation. One: because you release more often, you release a smaller set of changes. Which we think is going to reduce the risk. The second and more important for, maybe from an Agile point of view is that you get your feedback quicker.'—Team Lead*

Table 2. Risk Management and Regulatory Compliance Practices at FinTechCo

Discussion

Where other organisations might rely on CoBit or other compliance frameworks (Bernroider and Ivanov 2011) to ensure information security, FinTechCo applied Agile software development methods to address the regulations defined by APRA (APRA 2013) and to mitigate development risks. In this section, we will discuss how FinTechCo managed to apply the Agile methods as risk management strategy with the help of the Iversen et al. (2004) framework and how their method covers all the different aspects of risk management: method or technology, people and process.

The Agile method, Extreme Programming paired up with continuous delivery, established the boundaries around the ways of working and created a risk management scaffolding, defining what were the acceptable ways of working in the organisation. Selecting a very specific method reduced ambiguity and offered clear, yet flexible guidelines for both the employees to follow and the regulators to observe.

By selecting the Agile methods as the primary risk management strategy and the method to mitigate risks, FinTechCo had shifted their view on the people risk. Rather than seeing the people of the organisation as the sources of the risks (Crossler et al. 2013), FinTechCo treated their employees as the risk mitigators. Pair programming practice fostered awareness on issues and methods within the development teams and kept each team member working on the same products both aware on the status of the development as well as the potential issues of the products. By emphasising pair programming and by extending it to cover all technical collaboration, the organisation constantly ensured that two or more people observed the code or changes and minimised the risk of errors (Beck 1999).

The finer details of the development process, another aspect of risk strategy, were addressed by the methods selection as well. By applying issue tracking systems and continuous delivery practice, FinTechCo was successfully merging the Agile practice with the regulatory compliance. Agile development tools, such as the ticketing software, were carefully chosen to support both the regulations and the development workflow. No undocumented piece of code was applied to the environments that would compromise the security of the systems.

When comparing the risk management strategy chosen by FinTechCo to the Iversen et al. (2004) framework, one can identify many similarities between the framework and the steps of Agile software development. When organisations follow an Agile methodology, they are by default addressing all five steps required for strategic risk resolution for software development. **Figure 1** summarises the three levels of risk strategy, the methods, people and process and presents the Agile development practices linked with each layer along with the Iversen et al. framework.

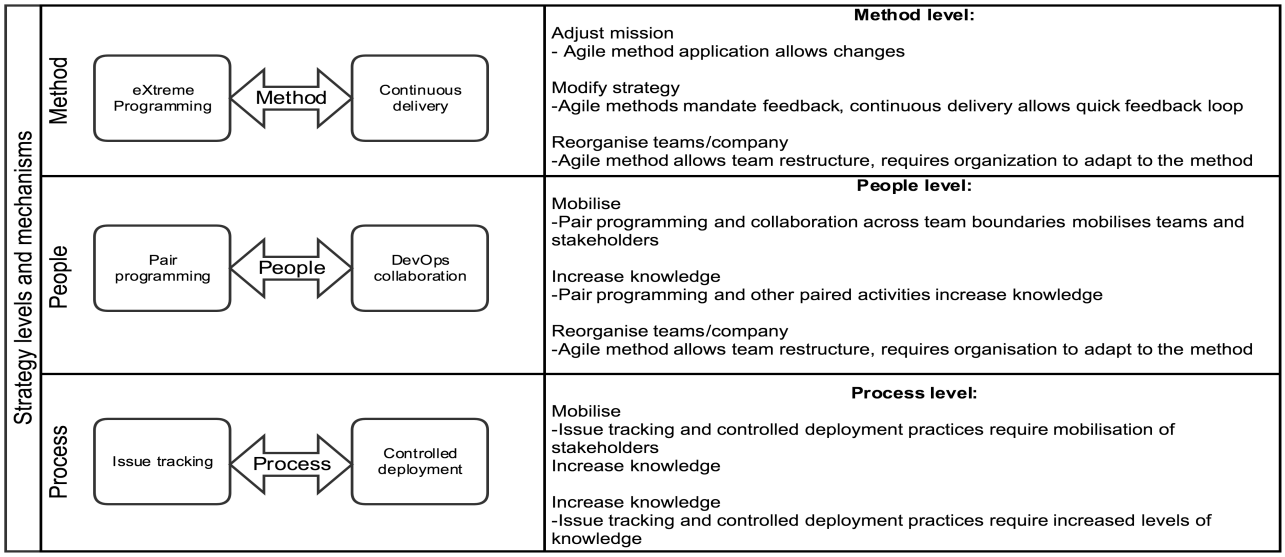

Figure 1. Risk strategy with Agile development

The first step, adjust the mission, is at the core of Agile software development method. Any project should welcome change at any point (Beck et al. 2001) and be ready to adjust the mission of the project. Similarly, the spirit of Agile is never opposed to strategy modifications, the second step, and by applying the constant delivery model; the projects can get early feedback on how the product is faring in the markets and adjust accordingly (Highsmith and Cockburn 2001). The third step, mobilizing the organisation is one of the goals of Agile development as well. A strong commitment is needed from the internal organisation stakeholders in the form of customer interactions or nominated product owners (Beck 1999). Agile development is, again, on the forefront of knowledge sharing and spreading tacit knowledge across the team members and other stakeholders (Highsmith 2002).

Finally, the last step of Iversen et al.'s framework suggests that the organisation might require reorganisations. We argue that organisations that apply Agile development methods are more open to reorganisation than plan-driven organisations and if the teams have the proper support from the upper management, the flat Agile organisation can reorganise to respond to changes better (Cockburn and Highsmith 2001).

These three aspects of strategy applied at FinTechCo are linked to the different aspects of strategy. Organisations that aspire to apply Agile development methods as their risk management strategy need to practise strategic hiring and ensure their employees are committed to the Agile methods to mitigate the human factors (Crossler et al. 2013). They also need to foster the Agile environment continuously and facilitate diligent application of the chosen Agile methods, strategically manage the human resources of the organisation from culture to change management (Grieves 2003). To ensure that the development process itself is not creating additional risks, the organisations need to continuously monitor and improve the application of the process, as one would in an Agile development environment (Highsmith 2002).

All these elements facilitate an environment that constantly allows assessment and mitigation of the risks (Iversen et al. 2004). We propose that these three levels of Agile method-based risk management can be extrapolated into an abstract model that describes the elements of risk management strategy via Agile development.

Figure 2 presents an illustration of our theory of Agile software development method as a risk-management strategy. The chosen Agile method, the strategic software development method, influences the strategic employee management, the people aspect of the risk strategy as well as the process aspects, the strategic work coordination. On the other hand, the chosen Agile method needs solid foundation of strategically managed people and process to function properly as a strategy for risk management. In our case study of FinTechCo, all three elements were strongly presented and were an integral part of the success of the organisation to both comply with regulations and succeed in the financial industry.

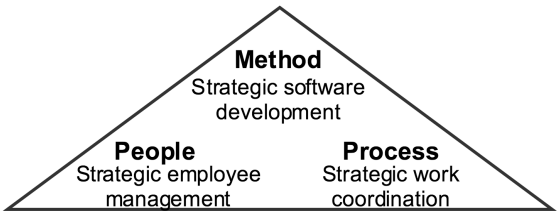

Figure 2. Agile software development as risk management strategy

Conclusion

This study presents a case study of a FinTech organisation which, by applying Agile methods, has been able to implement Agile software development methods as their primary risk management strategy. FinTechCo is a unique enterprise among organisations in the Australian finance sector. The case study provides an account of the successful operation, which has succeeded in implementing a challenging Agile framework in a highly regulated industry.

There are several implications for both practice and theory. From a practical perspective, our case study illustrated that an organisation can apply Agile development in the highly regulated and complex environment of finance. The case provided evidence for the applicability of the Extreme Programming method and details how such an approach can be beneficial when creating an environment of high performance. From a theoretical perspective, we have applied our findings from the case study to form a novel framework that illustrates the different factors that enable quality and assurance and derived a novel theory that links the aspects of Agile development methods with the elements of strategic risk management. As this is a qualitative study, we are not claiming that our single case study is generalisable across all cases but rather generalises from data to theory (Lee and Baskerville 2003).

The framework, albeit created from a rare case where XP was used, could be applied to analyse other organisations who apply other Agile development methods as well. The framework looks into the quality assurance factors and the distinctive areas of the development and can be applied to research whether the organisations have quality assurance factors in place. Application of Agile methods as risk management strategy is not limited to XP. This research could encourage other Agile and non-Agile organisations to assess their current risk management strategies and strengthen their Agile development approach.

REFERENCES

APRA, A.P.R.A. 2013. "Prudential Practice Guide, Cpg 234–Management of Security Risk in Information and Information Technology," *APRA,[Online], available at: http://www.apra.gov.au/CrossIndustry/Documents/ (accessed on 16th January 2018)*).

Barki, H., Rivard, S., and Talbot, J. 1993. "Toward an Assessment of Software Development Risk," *Journal of management information systems* (10:2), pp. 203-225.

Beck, K. 1999. "Embracing Change with Extreme Programming," *Computer* (32:10), pp. 70-77.

Beck, K. 2000. *Extreme Programming Explained: Embrace Change*. Addison-Wesley Professional.

Beck, K., Beedle, M., Van Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., and Jeffries, R. 2001. "Agile Manifesto, 2001." from http://www.agilemanifesto.org

Boehm, B.W. 1988. "A Spiral Model of Software Development and Enhancement," *Computer* (21:5), pp. 61-72.

Bernroider, E.W. and Ivanov, M., 2011. IT project management control and the Control Objectives for IT and related Technology (CobiT) framework. *International Journal of Project Management*, (29:3), pp.325-336.

Bulgurcu, B., Cavusoglu, H., and Benbasat, I. 2010. "Information Security Policy Compliance: An Empirical Study of Rationality-Based Beliefs and Information Security Awareness," *MIS quarterly* (34:3), pp. 523-548.

Chishti, S., and Barberis, J. 2016. *The Fintech Book: The Financial Technology Handbook for Investors, Entrepreneurs and Visionaries*. John Wiley & Sons.

Cockburn, A., and Williams, L. 2000. "The Costs and Benefits of Pair Programming," *Extreme programming examined*), pp. 223-247.

Crossler, R.E., Johnston, A.C., Lowry, P.B., Hu, Q., Warkentin, M., and Baskerville, R. 2013. "Future Directions for Behavioral Information Security Research," *computers & security* (32), pp. 90-101.

Dapp, T.F., Slomka, L., AG, D.B., and Hoffmann, R. 2014. "Fintech–the Digital (R) Evolution in the Financial Sector," *Deutsche Bank Research", Frankfurt am Main*.

de Lima Salge, C.A., and Berente, N. 2016. "Pair Programming Vs. Solo Programming: What Do We Know after 15 Years of Research?," *System Sciences (HICSS), 2016 49th Hawaii International Conference on*: IEEE, pp. 5398-5406.

Donaldson, S.E. and Siegel, S.G., 2001. *Successful software development*. Prentice Hall Professional.

Fitzgerald, B., Stol, K.-J., O'Sullivan, R., and O'Brien, D. 2013. "Scaling Agile Methods to Regulated Environments: An Industry Case Study," *Software Engineering (ICSE), 2013 35th International Conference on*: IEEE, pp. 863-872.

Grieves, J. 2003. *Strategic Human Resource Development*. Sage.

Gulamhuseinwala, I., Bull, T., and Lewis, S. 2015. "Fintech Is Gaining Traction and Young, High-Income Users Are the Early Adopters," *The Journal of Financial Perspectives, Winter 2015 FinTech*. The EY Global Financial Services Institute

Guo, Y., and Liang, C. 2016. "Blockchain Application and Outlook in the Banking Industry," *Financial Innovation* (2:1), p. 24.

Harrison, S., and Jan, J. 2017. "Information Security Management and the Human Aspect in Organizations," *Information and Computer Security* (25:5), pp. 494-534.

Herath, T., and Rao, H.R. 2009. "Encouraging Information Security Behaviors in Organizations: Role of Penalties, Pressures and Perceived Effectiveness," *Decision Support Systems* (47:2), pp. 154-165.

Highsmith, J., and Cockburn, A. 2001. "Agile Software Development: The Business of Innovation," *Computer* (34:9), pp. 120-127.

Hu, Q., Dinev, T., Hart, P., and Cooke, D. 2012. "Managing Employee Compliance with Information Security Policies: The Critical Role of Top Management and Organizational Culture," *Decision Sciences* (43:4), pp. 615-660.

Humble, J., and Farley, D. 2010. *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation (Adobe Reader)*. Pearson Education.

Hüttermann, M. 2012. *Devops for Developers*. Apress.

Iversen, J.H., Mathiassen, L., and Nielsen, P.A. 2004. "Managing Risk in Software Process Improvement: An Action Research Approach," *MIS Quarterly*, pp. 395-433.

Kumar, A., Saxena, A., Suvarna, V., and Ravat, V. 2016. "Top 10 Trends in Banking in 2016." Capgemini, www. capgemini. com/banking.

Leong, C., Tan, B., Xiao, X., Tan, F.T.C., and Sun, Y. 2017. "Nurturing a Fintech Ecosystem: The Case of a Youth Microloan Startup in China," *International Journal of Information Management* (37:2), pp. 92-97.

Leong, C., Tan, B., Xiao, X., Tan, F.T.C., and Sun, Y. 2017. "Nurturing a Fintech Ecosystem: The Case of a Youth Microloan Startup in China," *International Journal of Information Management* (37:2), pp. 92-97.

Lyytinen, K., Mathiassen, L., and Ropponen, J. 1998. "Attention Shaping and Software Risk—A Categorical Analysis of Four Classical Risk Management Approaches," *Information System Research* (9:3), pp. 233-255.

Marks, N. 2010. "Defining Grc: Internal Auditors Need to Make Sure They Understand Grc before Reporting on It to Executive Management and the Board," *Internal Auditor* (67:1), pp. 25-27.

McDonald, T., and Morling, S. 2011. "The Australian Economy and the Global Downturn Part 1: Reasons for Resilience," *Economic Round-up*:2), p. 1.

McHugh, M., McCaffery, F., and Casey, V. 2012. "Barriers to Adopting Agile Practices When Developing Medical Device Software," *Software Process Improvement and Capability Determination*), pp. 141-147.

Phaphoom, N., Sillitti, A., and Succi, G. 2011. "Pair Programming and Software Defects–an Industrial Case Study," *International Conference on Agile Software Development*: Springer, pp. 208-222.

Puschmann, T. 2017. "Fintech," *Business & Information Systems Engineering* (59:1), pp. 69-76.

Racz, N., Weippl, E., and Seufert, A. 2010. "A Frame of Reference for Research of Integrated Governance, Risk and Compliance (Grc)," *Communications and multimedia security*: Springer, pp. 106-117.

Ropponen, J., and Lyytinen, K. 2000. "Components of Software Development Risk: How to Address Them? A Project Manager Survey," *IEEE transactions on software engineering* (26:2), pp. 98-112.

Seo, J.-H., and Park, E.-M. 2017. "A Study on Financing Security for Smartphones Using Text Mining," *Wireless Personal Communications*), pp. 1-19.

Sharma, A. 1997. "Professional as Agent: Knowledge Asymmetry in Agency Exchange," *Academy of Management review* (22:3), pp. 758-798.

Stewart, H., Stewart, H., Jürjens, J., and Jürjens, J. 2017. "Information Security Management and the Human Aspect in Organizations," *Information & Computer Security* (25:5), pp. 494-534.

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

VersionOne, T.R. 2017. "11th Annual State of Agile Survey," Technical report, Version One.

Walsham, G. 1995. "Interpretive Case Studies in Is Research: Nature and Method," *European Journal of information systems* (4:2), p. 74.

Walsham, G. 2006. "Doing Interpretive Research," *European journal of information systems* (15:3), pp. 320-330.