



Do Scaling Agile Frameworks Address Risk in Global Software Development? An Empirical Study



Assoc. Prof. Tony Clear Tony.clear@aut.ac.nz

Mihi

- Hāere mai, Haere Mai, Haere Mai.
- Tēnā koutou katoa.
- Ko Tony Clear taku ingoa.
- Nō Pōneke ahau.
- Ko Maungakiekie taku maunga.
- Ko Waitematā taku moana.
- I te taha o taku matua, no Enniscorthy Ireland ahau.
- I te taha o aku whaea, no Cork Ireland ahau.
- Ko Tainui raua ko Ngapuhi nga iwi o nga mokopuna
- Tēnā koutou, Tēnā koutou, Tēnā tatou katoa.



Profile and Current Activities

- 1. Co-Director of Software Engineering Research lab at AUT https://serl.aut.ac.nz/
- 2. Global Software Engineering
 - 1. Scaled Agile
 - 2. Software Ecosystems RSNZ Catalyst Leaders with Prof Daniela Damian
 - 3. GSE Education
 - 4. Global Collaboration

3. Computing Education

- 1. Curriculum & Competencies [CC2020]
- 2. Onboarding Software Professionals SIGCSE Special Project [SERL]
- 3. Editorial Roles [ACM TOCE, ACM Inroads, Computer Science Education], PC various conferences CS Ed and SE
- 4. Teaching BCIS Papers Contemporary Issues in Software Engineering, Computing Technology In Society, R&D Project, B Eng(Hons)Industrial Project, plus Masters & PHD Supervision





Do Scaling Agile Frameworks Address Global Software Development Risks? An Empirical Study

Sarah Beecham, Tony Clear, Ramesh Lal, John Noll Journal of Systems and Software, Jan/Feb 2021 Journal First Int'l Conference on Global Software Engineering (ICGSE) 2021

> Presentation 19th May 2021 Tony Clear, Sarah Beecham, Ramesh Lal, John Noll



















Overview

- Focus of the paper
- Phase one Developing a GSD Risk Catalog
- Phase Two Theoretical Mapping
- Phase Three Empirical Assessment
- Conclusions



Paper Focus: Scaling Agile Frameworks and GSD Risks

- A three-phase process (lit review/practice-risk mapping/empirical validation),
- Illustrated how two scaling agile frameworks
 - –DAD and SAFe–
- Largely address 63 software development risks
- Identified in a GSD Risk Catalog
- But stronger in some areas than others



Phase One – Developing a GSD Risk Catalog

First phase:

- Identifying Global Software Development risks faced by software development organizations,
- By examining the literature on risks
 In both conventional and GSD contexts

Result:

- A GSD Risk Catalog of 63 risks,
- Divided into four quadrants following Wallace and Keil (2004):
 - 1. Customer Mandate,
 - 2. Scope and Requirements,
 - 3. Execution, and
 - 4. Environment



GSD Risk Catalog

Wallace, L., & Keil, M. (2004). Software project risks and their effect on outcomes. *Communications of the ACM,* 47(4), 68-73.

Quadrant 1: Customer Mandate (#8) Quadrant 2: Scope and Requirements (#10) 1.1 Conflict between users 2.1 Conflicting system requirements 1.2 Lack of cooperation from users 2.2 Continually changing project scope/objectives Lack of top management support for the 2.3 Continually changing system requirements project 2.4 Difficulty in defining the inputs and outputs of the 1.4 Lack of user participation system 2.5 Ill-defined project goals Lack or loss of organizational commitment to 2.6 Incorrect system requirements 1.6 Users not committed to the project 2.7 System requirements not adequately identified 1.7 Users resistant to change 2.8 Unclear system requirements 2.9 Undefined project success criteria 1.8 Users with negative attitudes toward the 2.10 Users lack understanding of system capabilities and risk Quadrant 4: Environment (#7) + 7 new GSD Quadrant 3: Execution (#28) + 3 new GSD risks ₽ 3.1 Development team unfamiliar with selected development tools 4.1 Change in organizational management during Perceived relative importance 3.2 Frequent conflicts among development team members Frequent turnover within the project team Corporate politics with negative effect on High level of technical complexity project Highly complex task being automated 4.3 Dependency on outside suppliers Immature technology 4.4 Many external suppliers involved in the 3.7 Inadequate estimation of project budget development project Inadequate estimation of project schedule 4.5 Organisation undergoing restructuring during Inadequate estimation of required resources 3.10 Inadequately trained development team members Resources shifted from the project due to 3.11 Ineffective communication changes in organizational priorities 3.12 Ineffective project manager 4.7 Unstable organizational environment 3.13 Inexperienced project manager 4.8 new: Country-specific regulations 3.14 Inexperienced team members 4.9 new: Delays caused by global distance 3.15 Lack of an effective project management methodology 4.10 new: Lack of architecture-organization MODERATE 3.16 Lack of commitment to the project among development team members 4.11 new: Lack of face-to-face interaction inhibits 3.17 Lack of people skills in project leadership knowledge sharing 3.18 Large number of links to other systems required 4.12 new: Lack of process alignment 3.19 Negative attitudes by development team 4.13 new: Lack of tool/infrastructure alignment 3.20 One of the largest projects attempted by the 4.14 new: Unstable country/regional organization political/economic environment 3.21 Poor project planning 3.22 Project affects a large number of user departments or 3.23 Project involves the use of new technology 3.24 Project involves use of technology that has not been used in prior projects 3.25 Project milestones not clearly defined 3.26 Project progress not monitored closely enough 3.27 Team members lack specialized skills required by the project 3.28 Team members not familiar with the task(s) being 3.29 new: Ineffective collaboration 3.30 new: Ineffective coordination 3.31 new: Lack of trust LOW HIGH Perceived Level of Control

Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information* and Software Technology, *56*(1), 54-78.

Verner, J. M.,

Fig. 2. GSD Risk Catalog derived from Wallace and Keil (2004) and Verner et al. (2014).

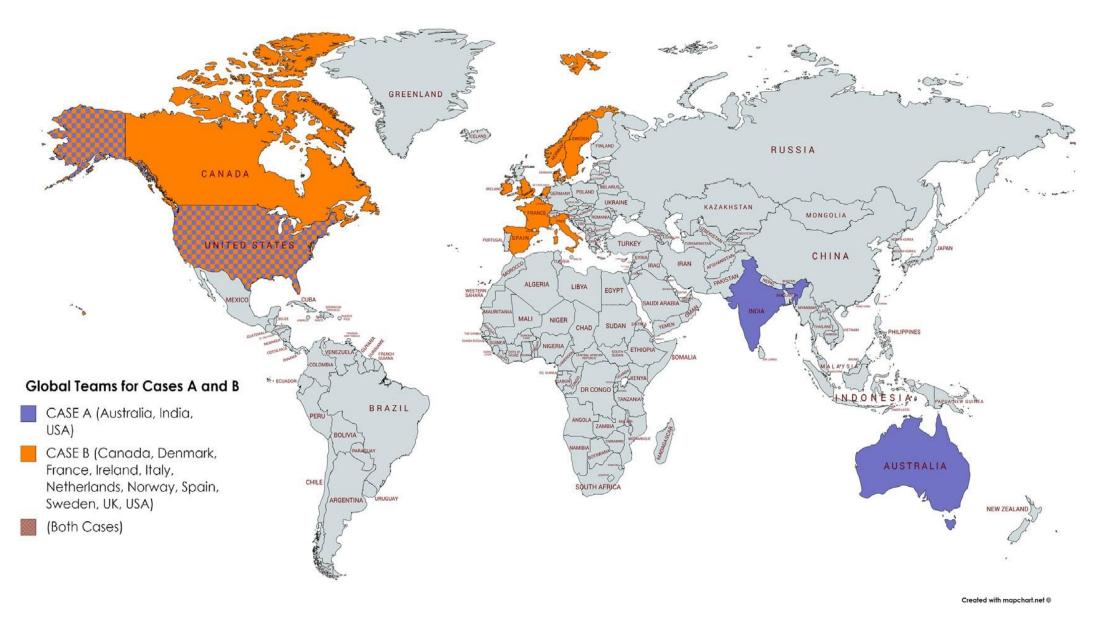
Phase Two - Theoretical Mapping

- Identified potential risk mitigation and elimination practices in the two scaling agile frameworks (DAD and SAFe).
- We compared these extracted practices (from DAD and SAFe) to risks in the GSD Risk Catalog (developed in Phase one)
- mapped practices to risks indicates how scaling agile practices might eliminate or mitigate those risks

Phase Three – Empirical Assessment

- Assessed the strength of the scaling agile frameworks to mitigate or eliminate risk,
 Avoiding criticism that we "speculated that the strategy would have helped observed problems" (Verner et al., 2014),
- Performed an empirical assessment of the theoretical mappings from Phase 2.
- To determine:
 - frequency with which practices in each framework performed in two companies,
 - and the **risks encountered** by those companies.

2 Global Cases and Locations



Phase Three – Empirical Assessment (Cont'd)

- Examined observation and interview notes and transcripts, self-assessment survey results,
- In multiple case study of two global software companies
- To understand
 - Extent to which DAD and SAFe practices (from theoretical mapping) were implemented in the company
 - Evidence of any of the risks (in the mapping or GSD Risk Catalogue) occurring in either company

If **practice is implemented** in a company and



risk NOT seen = theory supported

risk seen = theory unsupported



Phase Three – Broad Conclusion

Adds to **limited** empirical **evidence** of **efficacy** of **scaling agile frameworks**.

suggesting claims

Scaling Agile Frameworks address **risk** and are driven by **value**

have some validity!



Conclusions From Paper

Of the four quadrants in the GSD Risk Catalog

- Customer Mandate risks quadrant appears to be better addressed through the SAFe framework than DAD
- 2. Scope and Requirements risks are addressed well by both methods
- 3. Execution risks are better mitigated by DAD than SAFe
- 4. Environment risks are less well addressed by either approach

Suggests Environment set of risks are less amenable to being addressed by a process framework.



GSD Risks – not fully addressed by Scaling Agile Practices?

17 (out of 63) risks observed in both cases, eight are GSD risks specific:

- 1. Ineffective collaboration,
- 2. Ineffective coordination,
- 3. Lack of trust,
- 4. Country-specific regulations,
- 5. Delays caused by global distance,
- 6. Lack of architecture-organization alignment,
- 7. Lack of face-to-face interaction inhibits knowledge sharing,
- 8. Lack of process alignment.

Implications for all remote and home working?



Further Conclusions From Paper

Creating the GSD Risk Catalog

- Found many Global Software Development risks, not identified in the Wallace and Keil inventory
- Ten new risks GSD risks added to inventory
- Eight of these ten GSD risks observed (as shown on previous slide) in both companies, except:
 - Lack of tool/infrastructure alignment and
 - Unstable country/regional political/economic environment

These **new risks** appear to be **endemic** and suggest

a risk tariff in GSD

Final Takeaway

The **result** of a three phased methodology created a scaling agile risk theoretical mapping applied in a multiple case study

showing how two scaling agile frameworks

Disciplined Agile Delivery and the Scaled Agile Framework can potentially eliminate or mitigate the majority of

'software project' risks

- many global software development risks still pervade.

Scaling Agile Frameworks do not support every GSD risk.



Expansion



- GSD Risk Catalog
 - Process of dev't elaborated in Supplementary Technical Report pp. 5 7
 - Beecham, S., Clear, T., Lal, R., & Noll, J. (2020). *Companion to manuscript: Do Scaling Agile Frameworks Address Risk in Global Software Development? An Empirical Study* https://www.lero.ie/sites/default/files/Beecham 2020 TR003.pdf
- Practices by Framework
 - The result of these three phases is a **scaling agile risk theoretical mapping** that shows how two scaling agile frameworks— Disciplined Agile Delivery and the Scaled Agile Framework— can potentially eliminate or mitigate software project risks in global software development. [p. 21 also process elaborated on p.23 of TR]
 - Beecham, S., Clear, T., Lal, R., & Noll, J. (2021). Do Scaling Agile Frameworks Address Risk in Global Software Development? An Empirical Study. *Journal of Systems and Software*, 171(110823). https://doi.org/https://doi.org/10.1016/j.jss.2020.110823
 - Table 11 of paper DAD p.22
 - Table 12 of paper SAFE p. 25



Utility and Impact?

- BCIS Current R&D Project
- Security Policy for a Globally Distributed SME [Various Asia/Pacific sites]
 - Paananen, H., Lapke, M., & Siponen, M. (2020). State of the art in information security policy development. *Computers & Security, 88*, 101608.
 - Beecham, S., Clear, T., Lal, R., & Noll, J. (2021). Do Scaling Agile Frameworks
 Address Risk in Global Software Development? An Empirical Study. *Journal of Systems and Software*, 171(110823).
 - https://doi.org/https://doi.org/10.1016/j.jss.2020.110823
- Startup Software Dev't Org
- Markets a Software Solution which helps MSPs (Managed Service Providers) to automate common processes associated with IT support and administration



Case context?

- Culture of company
- Freewheeling style
- Historical flow of contract staff
- Inconsistent and Individualistic practices
- Personal use of open-source tools
- Security risks to be analysed and addressed
- Move towards more team-based development
- Desire to grow and scale
- How to develop and implement a security policy?
- Standardization, prioritizing, what to mandate vs. encourage?



Framing the situation?

- How to develop and implement a security policy?
- Standardization, prioritizing, what to mandate vs. encourage?

- Possible contribution from our scaled agile risk model?
- Focus on risks to determine priorities?
- Focus on practices to identify areas of concentration and sequence of activities?
- Potential contribution to framing both the problem and solution
- Issues highlighted from GSD Risk Catalog below

GSD Risk Catalog

Wallace, L., & Keil, M. (2004). Software project risks and their effect on outcomes. *Communications of the ACM,* 47(4), 68-73.

Quadrant 1: Customer Mandate (#8) Quadrant 2: Scope and Requirements (#10) 1.1 Conflict between users 2.1 Conflicting system requirements Lack of cooperation from users 2.2 Continually changing project scope/objectives Lack of top management support for the 2.3 Continually changing system requirements project 2.4 Difficulty in defining the inputs and outputs of the 1.4 Lack of user participation system Lack or loss of organizational commitment to 2.5 Ill-defined project goals 2.6 Incorrect system requirements 1.6 Users not committed to the project 2.7 System requirements not adequately identified 1.7 Users resistant to change 2.8 Unclear system requirements 2.9 Undefined project success criteria 1.8 Users with negative attitudes toward the 2.10 Users lack understanding of system capabilities and risk Quadrant 4: Environment (#7) + 7 new GSD Quadrant 3: Execution (#28) + 3 new GSD risks ₽ 3.1 Development team unfamiliar with selected development tools 4.1 Change in organizational management during Perceived relative importance 3.2 Frequent conflicts among development team members Frequent turnover within the project team Corporate politics with negative effect on High level of technical complexity project Highly complex task being automated 4.3 Dependency on outside suppliers Immature technology Many external suppliers involved in the 3.7 Inadequate estimation of project budget development project Inadequate estimation of project schedule 4.5 Organisation undergoing restructuring during Inadequate estimation of required resources 3.10 Inadequately trained development team members sources shifted the project due to 3.11 Ineffective communication changes in organizations 3.12 Ineffective project manager Unstable organizational environment 3.13 Inexperienced project manager 4.8 new: Country-specific regulations 3.14 Inexperienced team members 4.9 new: Delays caused by global distance 3.15 Lack of an effective project management methodology 4.10 new: Lack of architecture-organization 3.16 Lack of commitment to the project among development team members 4.11 new: Lack of face-to-face interaction inhibits 3.17 Lack of people skills in project leadership knowledge sharing 3.18 Large number of links to other systems required 4.12 new: Lack of process alignment 3.19 Negative attitudes by development team 4.13 new: Lack of tool/infrastructure alignmen 3.20 One of the largest projects attempted by the 4.14 new: Unstable country/regional organization political/economic environment 3.21 Poor project planning 3.22 Project affects a large number of user departments or 3.23 Project involves the use of new technology 3.24 Project involves use of technology that has not been used in prior projects 3.25 Project milestones not clearly defined 3.26 Project progress not monitored closely enough 3.27 Team members lack specialized skills required by the project 3.28 Team members not familiar with the task(s) being 3.29 new: Ineffective collaboration 3.30 new: Ineffective coordination 3.31 new: Lack of trust LOW HIGH Perceived Level of Control

Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information* and Software Technology, *56*(1), 54-78.

GSD Risk Catalog

Wallace, L., & Keil, M. (2004). Software project risks and their effect on outcomes. *Communications of the ACM,* 47(4), 68-73.

Quadrant 1: Customer Mandate (#8) Quadrant 2: Scope and Requirements (#10) 1.1 Conflict between users 2.1 Conflicting system requirements Lack of cooperation from users 2.2 Continually changing project scope/objectives Lack of top management support for the 2.3 Continually changing system requirements project 2.4 Difficulty in defining the inputs and outputs of the HIGH 1.4 Lack of user participation system Lack or loss of organizational commitment to 2.5 Ill-defined project goals 2.6 Incorrect system requirements Users not committed to the project 2.7 System requirements not adequately identified 1.7 Users resistant to change 2.8 Unclear system requirements 2.9 Undefined project success criteria 1.8 Users with negative attitudes toward the 2.10 Users lack understanding of system capabilities and risk Quadrant 4: Environment (#7) + 7 new GSD Quadrant 3: Execution (#28) + 3 new GSD risks ₽ 3.1 Development team unfamiliar with selected development tools 4.1 Change in organizational management during Perceived relative importance Frequent conflicts among development team members Frequent turnover within the project team Corporate politics with negative effect on High level of technical complexity project Highly complex task being automated 4.3 Dependency on outside suppliers Immature technology 4.4 Many external suppliers involved in the 3.7 Inadequate estimation of project budget development project Inadequate estimation of project schedule 4.5 Organisation undergoing restructuring during Inadequate estimation of required resources 3.10 Inadequately trained development team members Resources shifted from the project due to 3.11 Ineffective communication changes in organizational priorities 3.12 Ineffective project manager 4.7 Unstable organizational environment 3.13 Inexperienced project manager 4.8 new: Country-specific regulations 3.14 Inexperienced team members 4.9 new: Delays caused by global distance 3.15 Lack of an effective project management methodology 4.10 new: Lack of architecture-organization MODERATE 3.16 Lack of commitment to the project among development team members 4.11 new: Lack of face-to-face interaction inhibits 3.17 Lack of people skills in project leadership knowledge sharing 3.18 Large number of links to other systems required 4.12 new: Lack of process alignment 3.19 Negative attitudes by development team 4.13 new: Lack of tool/infrastructure alignment 4.14 new: Unstable country/regional 3.20 One of the largest projects attempted by the organization political/economic environment 3.21 Poor project planning 3.22 Project affects a large number of user departments or 3.23 Project involves the use of new technology 3.24 Project involves use of technology that has not been used in prior projects 3.25 Project milestones not clearly defined monitored closely enough Team members lacks lized skills required by the project 3.28 Team members not familiar with e task(s) being 3.29 new: Ineffective collaboration 3.30 new: Ineffective coordination 3.31 new: Lack of trust LOW HIGH Perceived Level of Com

Verner, J. M., Brereton, O. P., Kitchenham, B. A., Turner, M., & Niazi, M. (2014). Risks and risk mitigation in global software development: A tertiary study. *Information* and Software Technology, *56*(1), 54-78.



AUT SOFTWARE ENGINEERING Utility and Academic Impact? RESEARCH LABORATORY

Google Scholar Citations [19/05/2022 – 2021 paper]

Evaluating the role of scrum methodology for risk management in information technology enterprises Adoption of Large-Scale Scrum Practices through the Use of Management 3.0 Agile transformation, from classical-to agile project management in a multidisciplinary production environment, a case study Tools Engineers Need to Minimize Risk around CI/CD Pipelines in the Cloud Servitization Program Management Process Based On Scaled Agility: A Facilitating Framework 6 Analyzing SAFe Practices with Respect to Quality Requirements: Findings from a Qualitative Study Customer Controlled Managed Services Processes for Productivity Gains without Breaking Contractual Obligations: An Exploratory Study **Evaluating AGILE adoption** in software delivery organizations 9 Analyzing SAFe Practices with Respect to Quality Requirements: Findings from a Qualitative Study Toward Unveiling How SAFe Framework Supports Agile in Global Software Development An effective agile development process by a hybrid intelligent effort estimation protocol Agile and generic work values of British vs Indian IT workers: a culture-clash case Systematic Literature Review: Causes of Rework in GSD 14) Från kaffeautomaten till digitala mötesrum: Mjukvaruutvecklares upplevelser av att arbeta med agila utvecklingsmetoder under rådande pandemic From the coffee machine to digital meeting rooms: Software developers' experiences of working with agile development methods during the current pandemic

15)

Modelo de evaluación de metodologías de desarrollo de software web *Evaluation model of web software* development methodologies



Nga mihi

And

• Questions?

