Embedding Software Engineering in Mixed Methods: Computationally Enhanced Risk Communication

Ann Marie Reinhold
Montana State University

Madison H. Munro
Montana State University

Elizabeth A. Shanahan
Montana State University

Ross J. Gore
Old Dominion University, rgore@odu.edu

Barry C. Ezell
Old Dominion University, bezell@odu.edu

See next page for additional authors

Original Publication Citation

Reinhold, A. M., Munro, M. H., Shanahan, E. A., Gore, R. J., Ezell, B. C., & Izurieta, C. (2024). *Embedding software engineering in mixed methods: Computationally enhanced risk communication*. Zenodo. https://doi.org/10.5281/zenodo.10519749

Embedding Software Engineering in Mixed Methods: Computationally Enhanced Risk Communication

Ann Marie Reinhold[1], Madison H. Munro[1], Elizabeth A. Shanahan[2], Ross J. Gore[3],

Barry C. Ezell[3], and Clemente I. Izurieta[1,4,5]

Author Note

[1] Gianforte School of Computing, Montana State University

[2] Department of Political Science, Montana State University

[3] Virginia Modeling, Analysis and Simulation Center, Old Dominion University

[4] Pacific Northwest National Laboratory Joint Appointment

[5] Idaho National Laboratory Joint Appointment

We have no known conflict of interest to disclose.

Ann Marie Reinhold https://orcid.org/0000-0003-0411-3486

Elizabeth A. Shanahan https://orcid.org/0000-0002-1074-5570

Ross J. Gore https://orcid.org/0000-0003-4065-6146

Barry C. Ezell https://orcid.org/0000-0003-4274-908X

Clemente I. Izurieta https://orcid.org/0000-0002-1002-3906

Correspondence regarding this article should be addressed to Dr. Ann Marie Reinhold, P.O. Box 173880, Montana State University, Bozeman, MT 59717-3880.

Email: reinhold@montana.edu

Abstract

Mixed methods research ameliorates many convergent research challenges within the contemporary sociotechnical landscape. We suggest the integration of software engineering in mixed methods studies is a critical step to address some of the remaining and persistent challenges. One such research challenge where software engineering is particularly well suited is in hazard preparedness—in particular, the creation of risk communication messages to mitigate or prevent harm. Computationally enhanced risk communication is convergent research that integrates software engineering and social science research for the benefit of protecting humans and infrastructure. To this end, we developed a mixed methods framework for the efficient construction of risk communication messages. We call this the Domain Agnostic Risk Communication (DARC) framework and present it here. The DARC framework formalizes connections between software engineering and social science methods. It incorporates the best available science in risk communication research and a cadre of natural language processing techniques to impart validity, reliability, and precision into resultant messages. The DARC framework is highly modular owing to the incorporation of the software engineering principles of abstraction, extensibility, and encapsulation. While the focus of this position paper is on risk communication, we encourage the incorporation of software engineering into mixed methods research and the incorporation of mixed methods more broadly into software engineering experimentation.

Keywords: mixed methods, software engineering, risk communication, natural language processing, Domain Agnostic Risk Communication Framework

Embedding Software Engineering in Mixed Methods: Computationally Enhanced Risk Communication

Mixed methods research exists within a complex sociotechnical landscape (Geels, 2004), wherein interdependencies are prevalent among the social aspects of people and technical aspects of engineered artifacts, such as software (Cooper & Foster, 1971). In the current landscape, the use of software is nearly ubiquitous. Researchers are embedding software into mixed methods in uncharted, interesting, and interdisciplinary ways (González Canché, 2023; Moorkens, 2015; Nelson et al., 2021; Reinhold et al., 2023). However, direct reference to software engineering principles and standards is rare in mixed methods studies. Few papers that called on software engineering principles and standards directly to improve mixed methods protocols or procedures (but see Easterbrook et al. (2008)'s related work). Our position is that this is both a problem and an opportunity.

Incorporation of software in a mixed methods study can be problematic when the tenets of software engineering are not considered. As stated by Nelson, "A general lack of standardized guidelines and training around computer-assisted text analysis in sociology is producing a risky situation for the potential haphazard and undisciplined use of text analysis methods" (2020). In the language used by software engineers, Nelson asserts that end users would benefit from improved software "quality in use" (QIU). That is, they need software that has been validated *in the context in which it is used*. Such validation can be guided by software engineering standards such as the new ISO/IEC-25019:2023 Quality in Use Standard[1]. Like much of software engineering, assessing QIU hinges on integrating qualitative and quantitative information about software.

[1] https://webstore.iec.ch/publication/90024

At their core, mixed methods and software engineering are similar. Mixed methods combine "elements of qualitative and quantitative research approaches…for the broad purposes of breadth and depth of understanding and corroboration" (Johnson et al., 2007). Similarly, software engineering inherently blends creative innovation (e.g., qualitative design) with technical rigor (e.g., quantitative unit testing) (Petrillo et al., 2016; Seaman, 2008). In this paper, we draw on these similarities and assert that incorporating software engineering into mixed methods offers a promising avenue for achieving convergent research goals. One such convergent goal is to research the most effective means to communicate risk information to motivate message recipients towards protective actions.

Why Focus on Risk Communication?

Across hazard domains from cybersecurity to natural hazards, effective risk communication saves lives and money by motivating people to take protective actions before disaster strikes. Yet, a conventional risk communication generally takes a factual approach to explaining hazard information and unfortunately fails to engage target audiences in adopting risk reduction behaviors (e.g., patching critical software vulnerabilities, purchasing flood insurance). However, a narrative-based risk communication (Raile et al., 2022) deploys the "science of stories" (Jones et al., 2014; Shanahan et al., 2018) to effectively persuade humans to take protective actions.

Using the Narrative Policy Framework (Shanahan et al., 2018) as a theoretical foundation to build a replicable structure to risk messages has been a step toward systemization, but challenges remain given the subjective nature of language choice in the construction of risk communication messages. The *de facto* approach to constructing messages occurs in a proverbial black box, guided by expert opinion, without the rigor or benefits of software engineering

(Reinhold et al., 2023). A rigorous and replicable procedure to guide the construction of risk communication messages is still lacking. To use the language of software engineering, the QIU of the procedures used to create and assess risk communication messages is limited—meaning that studies employing these messages have numerous unmitigated threats to validity and reliability (Reinhold et al., 2023).

To address this gap, Reinhold and colleagues developed the “Persuasion with Precision Procedure” (PPP) (2023). The PPP is the first to apply natural language processing (NLP) to the *construction* of messages–to systematically improve message precision and efficacy (Reinhold et al., 2023). To be clear, Reinhold and colleagues are far from the first to employ NLP in social sciences or risk communication research (2023). In social science research, the application of NLP and machine learning on analyzing long-form texts, such as reports and responses to interview questions, is well established (Guetterman et al., 2018; O'Halloran et al., 2018; Rohrer et al., 2017). However, they are the first to embed NLP into a mixed methods procedure for the construction of risk communication messages.

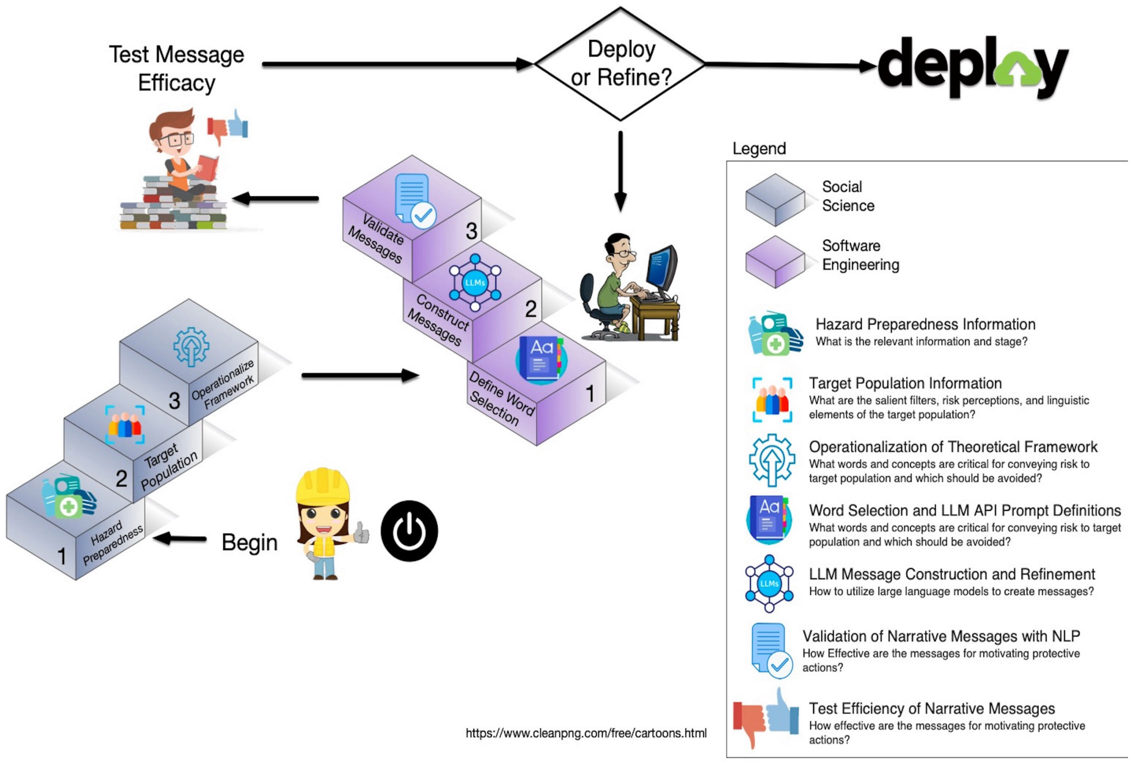
Reinhold and colleagues (2023) incorporation of NLP into the PPP is a paradigm shift for risk communication research because it solves the problem of off-the-cuff message construction. The PPP incorporates a cadre of NLP techniques, machine learning for word classification, and an algorithm to construct the messages. Resulting messages have demonstrable improvements in validity, reliability, and precision (Reinhold et al., 2023). Moreover, the investigation of the efficacy of these messages on natural hazard preparedness finds that the narrative-based messages constructed with this procedure were superior to conventional science messages (Raile et al., 2022; Shanahan et al., 2019).

Why Computationally Enhanced Risk Communication?

Message creation for hazard preparedness and crisis response often proceeds on a tight timeline, rendering the PPP of limited utility for practical applications. While the PPP was effective, it was also laborious and time consuming (Reinhold et al., 2023). The next frontier is the development of a framework for creating risk communication messages quickly without compromising the validity, reliability, and efficacy of resultant message. We approach this

Figure 1

Conceptual Overview of the DARC Framework



Hazard Preparedness Information
What is the relevant information and stage?

Target Population Information
What are the salient filters, risk perceptions, and linguistic elements of the target population?

Operationalization of Theoretical Framework
What words and concepts are critical for conveying risk to target population and which should be avoided?

Word Selection and LLM API Prompt Definitions
What words and concepts are critical for conveying risk to target population and which should be avoided?

LLM Message Construction and Refinement
How to utilize large language models to create messages?

Validation of Narrative Messages with NLP
How Effective are the messages for motivating protective actions?

Test Efficiency of Narrative Messages
How effective are the messages for motivating protective actions?

Note. The first three steps are rooted in the social sciences. The subsequent three steps are rooted in software engineering. Validated messages are tested and then either refined or deployed. LLM: large language model. API: application programming interface. NLP: natural language processing.

challenge with a mixed methods framework that formalizes the connections amongst software engineering and social science methods. We call this framework the Domain Agnostic Risk Communication (DARC) Framework.

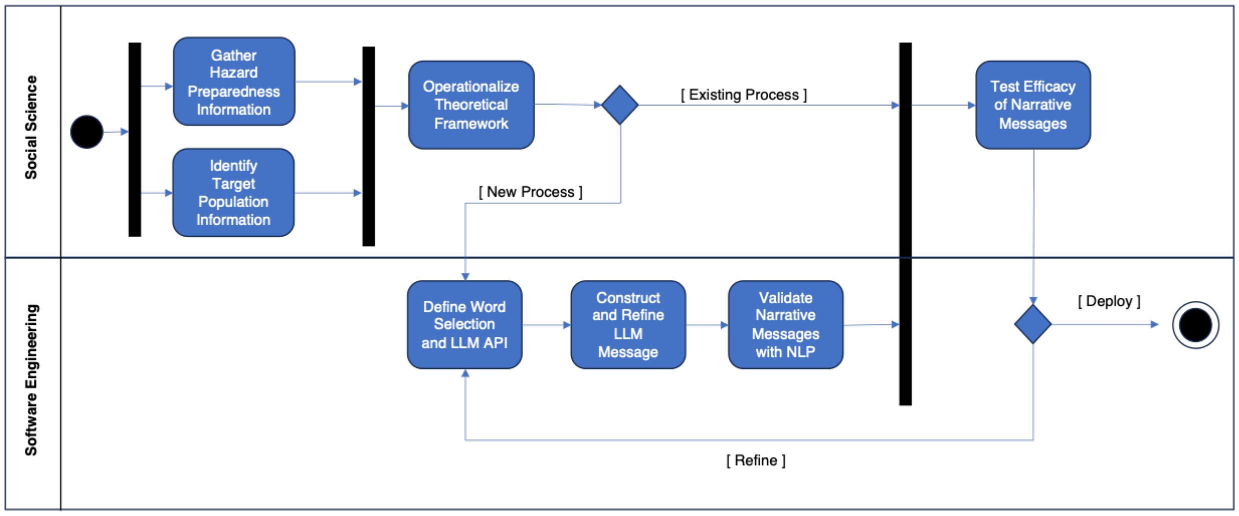
The DARC Framework builds from the PPP. The DARC Framework includes the components of the PPP that improve the operationalization of the Narrative Policy Framework and reduce threats to the validity and reliability in resultant messages. It also improves upon the PPP via the explicit incorporation of the software engineering principles of abstraction, extensibility, and encapsulation. Abstraction (Kramer, 2007) balances simplicity with efficacy to impart elegance in the design. Extensibility (Ingeno, 2018) ensures that the DARC Framework can be extended across hazard types and enhanced as necessary. Encapsulation (Lutowski, 2005) ensures that each component of the DARC Framework is modifiable by design; that is, the hazard domain, target population, theoretical framework, NLP algorithm(s), and large language model (LLM) are all interchangeable. Therefore, the DARC Framework takes the best of the PPP and improves upon it using software engineering tenets.

Here, we present the DARC framework as a series of steps in a conceptual diagram (Figure 1) and as a Unified Modelling Language (UML)[2] Activities diagram (Figure 2). UML is the current standard for presenting structural (e.g., components) and behavioral (e.g., processes) elements in software. The purpose for the dual presentations is as follows. One, it provides a plain illustration of how software engineering can be firmly embedded in a mixed methods procedure. Two, it clarifies that some of the steps are not sequence dependent (Social Science Steps 1 and 2 in Figure 1). Three, it demonstrates how the formalized presentation of UML is useful in presenting mixed methods frameworks and procedures.

[2] https://www.uml.org/

Figure 2

UML Activity Diagram Modelling the Flow of Activities in an Instantiation of the DARC Framework


Social Science
Software Engineering
Gather Hazard Preparedness Information
Identify Target Population Information
Operationalize Theoretical Framework
[Existing Process]
[New Process]
Test Efficacy of Narrative Messages
Define Word Selection and LLM API
Construct and Refine LLM Message
Validate Narrative Messages with NLP
[Deploy]
[Refine]

Note. Swimlanes indicate different actors, fork/join nodes (solid black lines) split or bring together potential parallel activities. Rhombuses are decision nodes. Solid black circle and circle with extra surrounding circle are start and stop nodes, respectively. LLM: large language model. API: application programming interface. NLP: natural language processing.

Prior to the starting node (Figure 2), the DARC framework requires that an appropriate theoretical framework is selected (e.g., Narrative Policy Framework), a target population is identified (i.e., the persons receiving the messages, such as cybersecurity dashboard watchers), and that the protective behavioral directives are known (e.g., “patch X vulnerability”). The starting node forks into parallel social science activity flows that include (1) the gathering and aggregation of hazard preparedness information, and (2) the identification and aggregation of critical information about the target population needed to develop persuasive messages. These flows join and are followed by the activity of operationalizing the theoretical framework, an activity that pertains to framing and constructing messages. The subsequent decision point proceeds with either the implementation of an existing process (e.g., a pre-trained algorithm) or creation of a new process. The creation of a new process involves stepping into the software engineering swimlane. The following activities proceed sequentially and involve (1) selecting words with the greatest persuasive power, (2) running the large language model (LLM) to create the risk communication message, and (3) validating the messages using NLP. At the subsequent join, the process is selected and the resultant messages are ready for testing. Results of testing indicate that messages are ready for deployment or require refinement. Once messages are deployed, the end of the activity flow is reached.

Discussion

Computationally enhanced risk communication is a convergent research challenge that integrates software engineering and social science research for the benefit of protecting humans and infrastructure. Our team is actively constructing messages with the DARC framework. While we have not yet reached a point where we are testing or deploying these messages, we are already finding the benefits of integrating software engineering explicitly into our mixed methods approach. Our experience thus far is that we have utilized abstraction, extensibility, and encapsulation effectively because the framework is proving applicable across hazard domains–ranging from natural hazards (flooding) to cybersecurity (phishing) to natural security (active shooter).

We created the DARC framework with QIU in mind, and this is proving beneficial. The modularity of the DARC framework facilitates versatility in how its components can be separated and recombined. This modularity provides wide context coverage across a range of use cases from risks ranging from natural hazards to national. For instance, one theoretical framework can be swapped for another just as easily as one LLM can be exchanged for another.

This plasticity exemplifies how powerful software engineering can be for mixed methods research. We can test a wide variety of use cases across hazard domains without compromising validity, reliability, or reusability.

The DARC framework is a representative example of convergent, mixed methods research that benefits from explicit incorporation of software engineering tenets. While its focus is on risk communication, we see great promise in the formal integration of software engineering into mixed methods research. The same software engineering principles that we employed here will be useful for solving other vexing challenges where convergence is imperative. Such challenges will require integrating activities across many scientific disciplines. Applying a software engineering point of view and utilizing software engineering tools (e.g., UML) can help address these challenges. The UML activities diagram (Figure 2) illustrates this point well. Adding another scientific discipline is as simple as adding an additional swimlane and delineating the interfaces amongst activities across the lanes. This flexibility may also make UML extremely useful for constructing joint displays.

The benefits of bridging across software engineering and mixed methods are not unidirectional. Here we demonstrate ways that direct integration of software engineering principles can improve mixed methods studies. However, mixed methods approaches can improve software engineering as well.

Although software engineering imported and adapted many methods from applied social science (Yin, 2008), social factors are not fully woven into the fabric of software engineering. For instance, the gold standard for training software engineers (Software Engineering Body of Knowledge [SWEBOK][3]) points towards the incorporation of important social factors, but does

[3] https://www.computer.org/education/bodies-of-knowledge/software-engineering/topics

not explore these factors in great depth (Glinz et al., 2023). This is problematic because, e.g., humans–as developers–create and mitigate technical debt (Avgeriou et al., 2016), just as humans–as end users–require user interfaces tailored to human needs (Afzal & Goues, 2018; Lin et al., 2017). In particular, experimentation in software engineering can benefit from mixed methods approaches that formally integrate qualitative and quantitative techniques, "where the emphasis is on using those methods that most effectively address the research problem" (Easterbrook et al., 2008). We conclude by reiterating enthusiasm for the incorporation of software engineering into mixed methods and vice versa—especially as interdisciplinary research continues to become the rule rather than the exception (Bachrach & Abeles, 2004; Leahey et al., 2019; Zuo & Zhao, 2018).

References

Afzal, A., & Goues, C. L. (2018). A study on the use of IDE features for debugging. *Proceedings of the 15th International Conference on Mining Software Repositories*, 114-117. https://doi.org/10.1145/3196398.3196468

Avgeriou, P., Kruchten, P., Ozkaya, I., & Seaman, C. (2016). Managing technical debt in software engineering (dagstuhl seminar 16162). *Dagstuhl reports*, *6*(4), 110-138. https://doi.org/10.4230/DagRep.6.4.110

Bachrach, C. A., & Abeles, R. P. (2004). Social science and health research: Growth at the National Institutes of Health. *American Journal of Public Health*, *94*(1), 22-28. https://doi.org/ https://doi.org/10.2105/AJPH.94.1.22

Cooper, R., & Foster, M. (1971). Sociotechnical Systems. *American Psychologist*, *26*(5), 467-474. https://doi.org/10.1037/h0031539

Easterbrook, S., Singer, J., Storey, M.-A., & Damian, D. (2008). Selecting Empirical Methods for Software Engineering Research. In F. Shull, J. Singer, & D. I. K. Sjøberg (Eds.), *Guide to Advanced Empirical Software Engineering* (pp. 285-311). Springer London. https://doi.org/10.1007/978-1-84800-044-5_11

Geels, F. W. (2004). From sectoral systems of innovation to socio-technical systems: Insights about dynamics and change from sociology and institutional theory. *Research Policy*, *33*(6), 897-920. https://doi.org/10.1016/j.respol.2004.01.015

Glinz, M., Seyff, N., Bühne, S., Franch, X., & Lauenroth, K. (2023). Towards a Modern Quality Framework. *2023 IEEE 31st International Requirements Engineering Conference Workshops (REW)*, 357-361. https://doi.org/10.1109/REW57809.2023.00067

González Canché, M. S. (2023). Graphical Retrieval and Analysis of Temporal Information Systems (GRATIS): An Integrative Mixed Methodology and Open-Access Software to Analyze the (Non-) Linear Chronological Evolution of Information Embedded in Textual/Qualitative Data. *Journal of Mixed Methods Research*, 15586898231166968. https://doi.org/10.1177/15586898231166968

Guetterman, T. C., Chang, T., DeJonckheere, M., Basu, T., Scruggs, E., & Vydiswaran, V. V. (2018). Augmenting qualitative text analysis with natural language processing: methodological study. *Journal of medical Internet research*, *20*(6), e231. https://doi.org/10.2196/jmir.9702

Ingeno, J. (2018). *Software Architect's Handbook* (P. Publishing, Ed.). https://www.packtpub.com/product/software-architects-handbook/9781788624060

Johnson, R. B., Onwuegbuzie, A. J., & Turner, L. A. (2007). Toward a definition of mixed methods research. *Journal of mixed methods research*, *1*(2), 112-133. https://doi.org/10.1177/1558689806298224

Jones, M., Shanahan, E., & McBeth, M. (2014). *The science of stories: Applications of the narrative policy framework in public policy analysis*. Springer. https://doi.org/10.1057/9781137485861

Kramer, J. (2007). Is abstraction the key to computing? *Commun. ACM*, *50*(4), 36–42. https://doi.org/10.1145/1232743.1232745

Leahey, E., Barringer, S. N., & Ring-Ramirez, M. (2019). Universities' structural commitment to interdisciplinary research. *Scientometrics*, *118*, 891-919. https://doi.org/10.1007/s11192-018-2992-3

Lin, Y., Sun, J., Xue, Y., Liu, Y., & Dong, J. (2017). Feedback-based debugging. *2017 IEEE/ACM 39th International Conference on Software Engineering (ICSE)*, 393-403. https://doi.org/10.1109/ICSE.2017.43

Lutowski, R. (2005). *Software Requirements: Encapsulation, Quality, and Reuse*. Taylor & Francis Group, LLC. https://doi.org/https://doi.org/10.1201/9781420031317

Moorkens, J. (2015). Consistency in Translation Memory Corpora: A mixed methods case study. *Journal of Mixed Methods Research*, *9*(1), 31-50. https://doi.org/10.1177/1558689813508226

Nelson, L. K. (2020). Computational grounded theory: A methodological framework. *Sociological Methods & Research*, *49*(1), 3-42. https://doi.org/10.1177/0049124117729703

Nelson, L. K., Burk, D., Knudsen, M., & McCall, L. (2021). The future of coding: A comparison of hand-coding and three types of computer-assisted text analysis methods. *Sociological Methods & Research*, *50*(1), 202-237. https://doi.org/10.1177/0049124118769114

O'Halloran, K. L., Tan, S., Pham, D.-S., Bateman, J., & Vande Moere, A. (2018). A digital mixed methods research design: Integrating multimodal analysis with data mining and information visualization for big data analytics. *Journal of Mixed Methods Research*, *12*(1), 11-30. https://doi.org/10.1177/1558689816651015

Petrillo, F., Soh, Z., Khomh, F., Pimenta, M., Freitas, C., & Guéhéneuc, Y.-G. (2016). Towards understanding interactive debugging. *2016 IEEE International Conference on Software Quality, Reliability and Security (QRS)*, 152-163. https://doi.org/10.1109/ICPC.2019.00030

Raile, E. D., Shanahan, E. A., Ready, R. C., McEvoy, J., Izurieta, C., Reinhold, A. M., Poole, G. C., Bergmann, N. T., & King, H. (2022). Narrative risk communication as a lingua franca for environmental hazard preparation. *Environmental Communication*, *16*(1), 108-124. https://doi.org/10.1080/17524032.2021.1966818

Reinhold, A. M., Raile, E. D., Izurieta, C., McEvoy, J., King, H. W., Poole, G. C., Ready, R. C., Bergmann, N. T., & Shanahan, E. A. (2023). Persuasion with precision: Using natural language processing to improve instrument fidelity for risk communication experimental treatments. *Journal of Mixed Methods Research*, *17*(4), 373-395. https://doi.org/10.1177/15586898221096934

Rohrer, J. M., Brümmer, M., Schmukle, S. C., Goebel, J., & Wagner, G. G. (2017). "What else are you worried about?"–Integrating textual responses into quantitative social science research. *PloS one*, *12*(7), 1-34. https://doi.org/10.1371/journal.pone.0182156

Seaman, C. B. (2008). Qualitative methods. In *Guide to advanced empirical software engineering* (pp. 35-62). Springer. https://doi.org/10.1007/978-1-84800-044-5_2

Shanahan, E. A., Jones, M. D., McBeth, M. K., & Radaelli, C. M. (2018). The narrative policy framework. In C. M. Weible (Ed.), *Theories of the Policy Process* (Fifth Edition ed., pp. 173-213). Routledge. https://doi.org/10.4324/9780429494284

Shanahan, E. A., Reinhold, A. M., Raile, E. D., Poole, G. C., Ready, R. C., Izurieta, C., McEvoy, J., Bergmann, N. T., & King, H. (2019). Characters matter: How narratives shape affective responses to risk communication. *PLoS One*, *14*(12), e0225968. https://doi.org/10.1371/journal.pone.0225968

Yin, R. K. (2008). *Case Study research: Design and Methods (Applied Social Research Methods)* (Fourth Edition ed.). Sage Publications.

Zuo, Z., & Zhao, K. (2018). The more multidisciplinary the better?–The prevalence and interdisciplinarity of research collaborations in multidisciplinary institutions. *Journal of Informetrics*, *12*(3), 736-756. https://doi.org/10.1016/j.joi.2018.06.006