Storytelling in human–centric software engineering research

Austen Rainer
a.rainer@qub.ac.uk
Queen's University Belfast
Belfast, County Antrim, Northern Ireland

ABSTRACT

BACKGROUND: Software engineering is a human activity. People naturally make sense of their activities and experience through storytelling. But storytelling does not appear to have been properly studied by software engineering research.
AIM: We explore the question: what contribution can storytelling make to human–centric software engineering research?
METHOD: We define concepts, identify types of story and their purposes, outcomes and effects, briefly review prior literature, identify several contributions and propose next steps.
RESULTS: Storytelling can, amongst other contributions, contribute to data collection, data analyses, ways of knowing, research outputs, interventions in practice, and advocacy, and can integrate with evidence and arguments. Like all methods, storytelling brings risks. These risks can be managed.
CONCLUSION: Storytelling provides a potential counter–balance to abstraction, and an approach to retain and honour human meaning in software engineering.

CCS CONCEPTS

• **Human-centered computing**; • **Applied computing → Law, social and behavioral sciences**; • **Software and its engineering**;

KEYWORDS

story, storytelling, narrative, qualitative inquiry, human–centric software engineering, behavioral software engineering, context, argument, evidence

ACM Reference Format:
Austen Rainer. 2021. Storytelling in human–centric software engineering research. In *Evaluation and Assessment in Software Engineering (EASE 2021), June 21–23, 2021, Trondheim, Norway*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3463274.3463803

∼ If story is central to human meaning why, in the research world, is there not more storytelling? [26]

1 INTRODUCTION

Software engineers generate and transform abstractions but they are not themselves abstractions. Software engineers have the same human characteristics as non–software engineers. They have goals [52] and intentions [14, 25]; they wrestle with motivation [4], happiness [15], stress [32], politics [5], ethics [44] and human values [55]. They work individually, and in teams, projects, organisations, the wider software industry, and society [9, 10]. The software engineer's work with abstractions therefore takes place within the common sphere of human activity and experience. In other words, software engineering (SE) is a human–centric activity [16, 25].

Haidt writes that the human mind is a story processor not a logic processor ([17], p. 328) and that, "…[a]mong the most important stories we know are stories about ourselves…" ([17], p. 328). Storytelling would therefore be a natural way for software engineers to make sense of their own and other's behaviour, and for researchers to better understand this human–centric activity.

Given the above, this paper explores the following simply–stated question: what contribution can storytelling make to human–centric software engineering research? The paper identifies several opportunities for storytelling to contribute to software engineering research, recognises there are risks (that can be managed) in using storytelling, and proposes next steps.

The remainder of this paper is organised as follows: in Section 2 we provide foundations, including concepts, challenges in SE research and a brief review of prior research; in Section 3 we discuss several ways in which storytelling can contribute to SE research; and in Section 4 we briefly consider next steps, and then conclude.

EASE 2021, June 21–23, 2021, Trondheim, Norway
© 2021 Copyright held by the owner/author(s).
ACM ISBN 978-1-4503-9053-8/21/06.
https://doi.org/10.1145/3463274.3463803

2 FOUNDATIONS

2.1 Concepts

There are many definitions for the terms *story*, *storytelling* and *narrative*, and disciplines use other terms too. For example, Shaffer et al. [39] observe that journalism uses the term *exemplar*, marketing uses *testimonial*, psychology uses *case study* or *case history*, sociologists use *personal stories* and medicine uses *narrative*. Polkinghorne [34] explores contrasting definitions of the term *narrative*, distinguishing between the following: narrative as a prosaic discourse differentiated from poetic discourse, narrative as the qualitative inquiry into naturally produced linguistic expressions in their context, narrative as the collected body of data for analysis, narrative as the research output from qualitative inquiry, and narrative as a special type of discourse production, i.e., the story. We use Polkinghorne's [34] definition/s of storytelling, summarised in Table 1.

It can be convenient to distinguish between story and storytelling, where story is the (static) output from a storytelling process. An exemplar of story is the novel. Although the story may in some sense be static, the story remains a dynamic interaction between the output and the reader. So although in some sense a printed story is fixed it is available as input to a re–telling process: the story*telling* continues in the reading of the story. Digressing briefly, this suggests an intriguing metaphor for software: software is static until it is read – 'retold' – by a processor.

Just as researchers and story–analysts do not agree on definitions for narrative and for story, so there is disagreement on the necessary elements of a story. Indicative elements of storytelling include: a plot, a narrative structure, one or more protagonists, one or more antagonists, conflict, inciting incident/s, locations in time/s and space/s, and 'space' for the reader to 'inhabit' the story. See, for example, [7, 47] for further information on such elements.

Shaffer and Zikmund–Fisher [40] identify several purposes and outcomes for storied narrative, and Shaffer et al. [39] identify nine effects of narratives. These purposes, outcomes and effects are summmarised in Table 2. As the table indicates, storytelling has many outcomes relevant to software engineering practice and research.

2.2 Challenges to software engineering

Like all disciplines, software engineering faces fundamental challenges. We very briefly consider some of those challenges here, emboldened in the following discussion. In Section 3, we suggest that storytelling can help us to make progress on at least some of these challenges.

Software engineering practice and research is **multidisciplinary**, integrating many different disciplines from social science through to discrete, and now quantum, mathematics. There is the challenge of respecting the contrasting ways of knowing and types of knowledge from these disciplines. Evidence Based Software Engineering (EBSE) aims "…to improve decision making related to software development and maintenance by integrating current best evidence from research with practical experience and human values" ([11], p. 59). To these we may add system constraints. These different elements – research, practical experience, human values and constraints – imply different kinds of knowledge, and there is the challenge of **integrating different kinds of knowledge**. The **context** of software practice undermines our ability to generalise findings from research. Context also introduces postmodernist perspectives into software engineering research, e.g., that there is no objective truth at the levels of reality relevant to software engineering. Practitioners and researchers value different kinds of **evidence**. Different kinds of evidence align with different ways of knowing and support different kinds of knowledge. This introduces the challenge of **persuading software practitioners to improve on the basis of research**. There is increasing **methodological diversity** bringing a challenge to evaluating the quality of research. Finally, there is the challenge of **retaining, even honouring, meaning** in software engineering in the face of abstraction. In two recent tweets, Michael "GeePaw" Hill writes the following: "The software trade has a great many problems, but among the most debilitating and dangerous is the steadfast refusal to adequately incorporate the humanity of the makers into its culture, organization, and reasoning." [19] and, "By a process of relentless abstraction and ruthless compartmentalization, we seek time and again to suppress, distract from, and minimize the most central fact of software development: humans make software." [20].

2.3 Story and storytelling in research

Storytelling is recognised as a legitimate focus of study in other scientific disciplines, e.g., energy and climate change research [29], law [2, 51], organisational research [53], and behavioural medicine [39]. And whilst storytelling is recognised in a variety of disciplines related to software engineering – e.g., information systems [38], human–computer interaction [6], computer supported cooperative work [31], information visualisation [13], multimedia systems [27] and computer science education [21, 30] – software engineering practice and research has tended to direct attention only at one particular type of *story*, i.e., the user story.

In behavioural medicine, Shaffer et al. [39] cite prior work to assert several benefits with narrative–as–story. But Shaffer et al. [39] also write that, "…interventions with narratives appear to escape the scrutiny that interventions with statistical evidence receive from people who disagree with the message. This has important implications for health behavior change interventions [and, for the current paper, there are important implications for interventions in software engineering practice], where the goal is often to change attitudes towards an unhealthy or harmful health behavior [or, for the current paper, the goal of improving a suboptimal or counter-productive software practice]." ([39], p. 431). We return to Shaffer et al.'s work [39, 40] later in this paper.

There is some attention in software engineering research directed at narrative analysis and synthesis (e.g., [8]) however that work uses the term *narrative* in the sense of the qualitative inquiry into and with texts rather than as *story*. We find several papers that explicitly recognise story and storytelling in software engineering research: Ahonen and Sihvonen's [1] story of software process improvement; Lenberg et al.'s unpublished manuscript [24] on guidelines for qualitative studies of behavioural software engineering; and several papers (e.g., [28, 41]) that study war stories using storytelling as a method of data collection. We consider these publications in the next section of this paper.

3 THE CONTRIBUTION OF STORYTELLING TO HUMAN–CENTRIC SE RESEARCH

We present several arguments below for why and how storytelling can contribute to software engineering research and practice.

3.1 Storytelling as collected data

Lutters and Seaman [28] developed a simple protocol for collecting war stories from SE practitioners. That protocol guided practitioners on how to tell their stories. Lutters and Seaman [28] therefore already demonstrate that software engineering research collects some kinds of story from software practitioners. The stories collected have been stories of exception, e.g., of something that has gone wrong. What constitutes an exception will vary from practitioner to practitioner, e.g., by definition an expert experiences different kinds of exception to a novice. Shaffer et al. [39] show that non–exceptional stories, e.g., of process, can also be valuable. Sim and Alspaugh [41] show that collecting war stories is not necessarily just about acquiring a text for analysis. So although stories have been collected in software engineering research, there is much greater opportunity for collecting stories than has been undertaken to date.

Table 1: Example descriptions of storied narrative (from Polkinghorne [34])

#	Description
1	"A story is a special type of discourse production. In a story, events and actions are drawn together into an organized whole by means of a plot. A plot is a type of conceptual scheme by which a contextual meaning of individual events can be displayed." ([34], p. 7)
2	"...a specific kind of prose text (the story) and...the particular kind of configuration that generates a story (emplotment)." (p. 5)
3	"A storied narrative is the linguistic form that preserves the complexity of human action with its interrelationship of temporal sequence, human motivation, chance happenings, and changing interpersonal and environmental contexts. In this context, *story* refers not only to fictional accounts but also to narratives describing 'ideal' life events such as biographies, autobiographies, histories, case studies, and reports of remembered episodes that have occurred." ([34], p. 7; emphasis in original), and
4	"The subject–matter of stories is human action. Stories are concerned with human attempts to progress to a solution, clarification, or unraveling of an incomplete situation. " ([34], p. 7)

Table 2: Purpose & outcome [40], & effect [39] of storytelling

#	Purpose	Outcome
a	To inform	Improved knowledge
		Improved affective forecasting
b	To engage	Greater engagement
		Greater transportation
		Greater time spent with materials
c	To model behaviour	Increased participation
		Increased shared decision making
		Altered behavioural intentions
		Increased uptake of behaviours
d	To persuade	Altered behavioural intentions
		Increased uptake of behaviours
e	To provide comfort	Reduced psychological distress
		Reduced anxiety
#	**Effect**	
Communicate information more effectively		
1	More engaging	
2	Better recall	
3	Develop fewer counter–arguments	
Change attitudes, judgements and behaviours		
4	Increase attitudes	
5	Reduce prejudice	
6	Promote positive behaviour	
7	Reduce negative behaviour	
8	Improve work performance	
9	Ignore base rate information and increase narrative information	

3.2 Storytelling as analysis

Sim and Alspaugh [41] show that software engineering research has, to date, constrained the way it analyses stories and therefore limited the opportunities for learning from stories. They demonstrate richer ways of analysing stories, drawing on the humanities. They provide example approaches that, due to space, we are unable to review here.

3.3 Storytelling and ways of knowing

There are many different classifications of knowledge and therefore of ways of knowing. We use Heron's [18] four ways of knowing, briefly summarised here in Table 3. For a detailed exploration of these four ways of knowing, see Heron's book, *Co–operative inquiry: research into the human condition* [18]. A research field such as software engineering research that is, or seeks to be, multidisciplinary (e.g., [43]) must be open to different kinds of knowledge, and therefore different ways of knowing, and not only open to different propositional knowledge.

3.3.1 Grounding ways of knowing. In Table 3, the four ways of knowing are ordered, with experiential knowing positioned 'lowest' and practical knowing positioned 'highest'. This positioning is because the higher–positioned ways of knowing are grounded in the lower–positioned ways of knowing. In software engineering research, for example, we 'ground' our propositional knowledge in empirical evidence, and we seek to ground recommendations to practitioners (cf. practical knowledge) in propositional knowledge.

3.3.2 Intuition and presentational knowing. Heron's description of presentational knowing as "...an intuitive grasp of the significance of patterns" might misleadingly suggest that presentational knowing is an unconscious or semi–conscious activity. Storytelling is a form of conscious, intentional presentational knowledge. This is of course most obvious with published novels.

3.3.3 Assumptions about the status of ways of knowing. Heron is challenging at least two assumptions, i.e., 1) that propositional knowledge should be the pre–eminent way of knowing, and 2) that propositional knowledge is self–sufficient. Sims et al. [41, 42] discuss methodical and amethodical knowledge, also challenging the pre–eminence of any particular kind of knowledge.

3.3.4 The status of experiential and presentational ways of knowing in SE research. It is not clear whether these two types of knowing are distinctly recognised in software engineering research. When we conduct interviews, surveys, focus groups etc. we appear to be accessing presentational ways of knowing. It is frequently hard to directly access the experiential knowledge of others.

Table 3: Heron's [18] four ways of knowing

Way	Brief description
Practical	'…knowing how to exercise a skill…" (p. 52)
Propositional	"…intellectual statements, both verbal and numeric, conceptually organized in ways that do not infringe the rules of logic and evidence. Propositional knowledge is regarded both as pre–eminent and self–sufficient." (p. 33)
Presentational	"…an intuitive grasp of the significance of patterns as expressed in graphic, plastic, moving, musical and verbal art–forms…" (p. 52), e.g., story as presentational knowing.
Experiential	"…imaging [not imagining] and feeling the presence of some energy, entity, person, place, process or thing." (p. 52)

3.4 Integrating storytelling with evidence and argument for evaluation and assessment

Anderson et al. [2] write: "…in factual enquiries…for a story to be accepted as true it needs to be warranted by (anchored in) evidence. A well–informed story needs to be coherent, but to be true, it must be both plausible and backed by particular evidence." ([2], p. 283)

Rainer [35] developed a preliminary methodology for identifying, extracting and analysing arguments, evidence and explanations from texts, and for presenting those in a structured, integrated way. One type of explanation was the story. He demonstrated the application of the methodology to several examples taken from a blog post by Joel Spolsky [46], these examples being 'micro–war stories'. The methodology seeks to address Anderson's position, e.g., to evaluate the story using evidence. The methodology is open to the limitation that Sim and Alspaugh [41] observed with Lutters and Seaman's [28] approach, i.e., it extracts facts and information from a text. There is therefore the opportunity to extend the methodology. For example, there are opportunities to use story to help evaluate and assess software engineering. Anderson [2] shows how storytelling can help identify gaps in evidence and can help generate hypotheses.

3.5 Storytelling as output

There appear to be very few examples where SE researchers publish their research output *as a story*. Ahonen and Sihvonen's [1] paper is the only complete example we can find. Ahonen and Sihvonen [1] present a real–world story over a 2.5yr period, told from the point of view of an individual software engineer, of a Software Process Improvement (SPI) effort. They write, "The story…is based mainly on the personal experiences of a single software engineer. Those experiences have been documented by the engineer…, but those experiences have [also] been checked by interviewing several other people…" and, "The reason why a story like this should be interesting for others is that the mistakes made in SPI efforts during the documented time are quite universal."

There are many examples of researchers presenting fragments of story, e.g., Sim and Alspaugh [28, 41]. One explanation for the limited recounting of stories is the amount of publication 'real estate' they require.

Table 4: Shaffer et al.'s types of story [39, 40]

Types	Effect
Outcome	Ability to persuade
	Change attitudes
	Alter intentions
	Alter behaviours
Process	Identify relevant decision attributes
	Model decision processes
	Change how people search for information
Experience	Reduce affective forecasting errors
	Facilitate more accurate perspective–taking
	Improve resilience

3.6 Storytelling as intervention

To explain how storytelling can act as an intervention we draw on Shaffer et al.'s [39] explanatory model. We choose this model because it has been developed by and for scientists.

Shaffer et al. [39] developed the Narrative Immersion Model (NIM) to better understand how storied narrative works so that storied narrative might be used to help behaviour change, e.g., in patients. The NIM may therefore be understood as a model to support intervention. Whilst an intervention might occur *after* the research has taken place it is often the case that we design for intervention, e.g., design our studies with the intention of using the results to improve software practice. One implication is that researchers consider storytelling as part of the design of the study.

Drawing on their prior work [40], Shaffer et al. [39] define three types of story (see Table 4). These types are not intended to be exclusive: an actual story might fit more than one type. *Outcome* stories describe how a situation ends. *Process* stories describe how a decision was made. *Experience* stories describe what a real–world situation was really like. Each type of story has a different effect on behaviour. Depending on the kind of effect the researchers seek (cf. Table 4), the researchers might design for different kinds of story.

Comparing Table 4 with Table 2, notice that the three types of story identified by Shaffer et al. [39] in Table 4 do not seem to have a direct impact on improving knowledge; or in other words, these stories do not seem to have the function to inform. This may be because Shaffer et al. have not had the opportunity to investigate this outcome. We make this observation because software engineering research often focuses on knowledge and, more specifically, on propositional knowledge. Storytelling does not seem to naturally fit that type of knowledge, but instead storytelling complements it.

As well as talking about types of story and their effects, Shaffer et al. [39] also talk about the magnitude of effect of a story. Shaffer et al. [39] propose a continuum through which a reader may 'travel' from Interest through Involvement to Immersion. For Shaffer et al., the deeper into the narrative a reader 'travels' the more powerful the narrative will be to influence behaviour. Characteristics of the narrative promote deeper involvement. These characteristics appear to align the NIM with elements of storytelling (cf. Section 2) and with advice on creative writing.

3.7 Storytelling as advocacy

Stories and storytelling provide a way to advocate for, and/or to raise awareness of, important issues; in other words, human values. Consider a software engineering team developing a software system for managing information about children in publicly–funded care homes. What insights might Sissay's [45] memoir, of his life in managed care, offer to that software engineering team to help ensure they develop a human–centric software system, and not simply develop an algorithmic bureaucracy? Or, as another example, what insights might Kafka's *The Trial* and *The Castle* provide on AI–based decision–making (e.g., [3])? Other examples include: Ford's recently published memoir, *Think Black* [12], about his father, John Stanley Ford, hired by Thomas J. Watson to become IBM's first black software engineer; Wiener's memoir, *Uncanny Valley* [54], describing her experiences working in technology companies in Silicon Valley; and Kim's two fictional accounts, one co–written with colleagues, about DevOps [23] and software development [22].

3.8 The risks of storytelling

Anderson et al. [2] write, "…story telling is vulnerable to abuse. It may be true that stories and storytelling are psychologically necessary to decision–making in legal contexts, but they are dangerous in that they often can be used to violate logical standards, appeal to emotion rather than reason, and subvert legal principles and conventions." ([2], p. 280). Anderson et al. [2] caution about the careful use of story but do not argue *against* the use of story. They list a number of examples of dangers with story. They also develop a 'rough working protocol' for assessing the plausability, coherence and evidentiary support for a story.

All research methods have strengths and weaknesses and the researcher seeks to increase their awareness of the threats to validity that arise with each method. One contribution of storytelling is to help researchers remain aware of threats. Such threats don't just occur in the telling of *stories*. For example, Sim and Alspaugh [41] observe that storytelling is performance: interviewees select and filter what and how they share information. Such selection and filtering doesn't just occur with the telling of a story, however. Any interview can be understood as a performance, with the threats to validity that might arise.

A highly–influential paper in software engineering research, Parnas' [33] *A rational design process: how and why to fake it*, explicitly recognises the value of 'faking' something: "We will never find a process that allows us to design software in a perfectly rational way. The good news is that we can *fake it*. We can *present* our system to others as *if we* had been rational designers and it pays to *pretend* [to] do so during development and maintenance. ([33], p. 251; emphasis added). Parnas explains the scope of the pretence and argues for its value in certain circumstances. In a fundamental way, abstraction is also pretence, for it presents a decontextualised and therefore simplified version of reality. We accept that pretence because there is value in doing so. We suggest that storytelling shouldn't be summarily dismissed on the basis of pretence.

4 DISCUSSION

4.1 Where next?

There are many aspects of storytelling in human–centric software engineering that we have not discussed in this paper and for which further research is needed. Areas requiring further attention include:

(1) A systematic review of storytelling that builds on the brief review presented in Section 2.3. Such a review should consider both the humanities and the sciences, as well as creative writing and 'story analysts', e.g., [7, 47].
(2) The development of conceptual framework/s for storytelling that build on the foundations introduced in Section 2. Included within this framework would be the development of appropriate terminology, e.g., *argued storytelling* or *evidence–based storytelling* are phrases that could communicate the intent of the kind of storytelling we consider here.
(3) The development of research methodology and methods that draw on work from other disciplines. Several examples have been discussed in this paper, e.g., [2, 28, 35, 39, 41, 50].
(4) Guidance on the use of methodology and on assessing the quality of storytelling research, cf. [36].
(5) Developing ways in which stories in SE can be evaluated and assessed, but also how storytelling might be used to evaluate and assess software engineering. For example, Section 3.4 discussed the integration of storytelling with argument and evidence, and also the use of storytelling to generate hypotheses, and Section 3.6 discussed the use of storytelling for intervention.
(6) Investigating the use of storytelling as an approach to knowledge exchange with industry, and as a method of appropriate intervention in practice, cf. [39].
(7) Investigating the benefits of a more sophisticated understanding of storytelling, beyond the user story, for software engineering practice.
(8) The use of both reading groups and writing groups for developing skills in appreciating stories in research and in practice. Section 3.7 presented examples that reading groups might use.
(9) The development of platforms, resources and (social) networks to disseminate work, including supplementary materials. Examples to draw on include StoryCorps [49], The Story Collider [48] and experience repositories [37].

4.2 Conclusion

In this paper, we position storytelling as a particular kind of narrative. We show that whilst narrative synthesis is recognised in software engineering, storytelling has not been properly considered. We explore types of story and the outcomes and effects of these different story types. We argue that storytelling has a valuable, even necessary, contribution to many areas of human–centric software engineering. We recognise that stories can be dangerous (e.g., because they can mislead) but also that these dangers might be assessed and mitigated. We suggest that storytelling may act as a potential counter–balance to abstraction, and a means to retain and honour meaning in software engineering.

ACKNOWLEDGMENTS

We thank Dr Maria Angela Ferrario for her comments on an early version of this paper, and the anonymous reviewers for their thoughtful and constructive feedback.

REFERENCES

[1] Jarmo J Ahonen and Hanna-Miina Sihvonen. 2005. How things should not be done: A real-world horror story of software engineering process improvement. In *European Conference on Software Process Improvement*. Springer, 59–70.

[2] Terence Anderson, David Schum, and William Twining. 2005. *Analysis of evidence.* Cambridge University Press.

[3] Torben Beck Jørgensen. 2012. Weber and Kafka: The rational and the enigmatic bureaucracy. *Public Administration* 90, 1 (2012), 194–210.

[4] Sarah Beecham, Nathan Baddoo, Tracy Hall, Hugh Robinson, and Helen Sharp. 2008. Motivation in Software Engineering: A systematic literature review. *Information and software technology* 50, 9-10 (2008), 860–878.

[5] Mark Bergman, John Leslie King, and Kalle Lyytinen. 2002. Large-scale requirements analysis revisited: the need for understanding the political ecology of requirements engineering. *Requirements Engineering* 7, 3 (2002), 152–171.

[6] Mark Blythe. 2017. Research fiction: storytelling, plot and design. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems.* 5400–5411.

[7] Shawn Coyne. 2015. *The story grid: what good editors know.* Black Irish Entertainment LLC.

[8] Daniela S Cruzes, Tore Dybå, Per Runeson, and Martin Höst. 2015. Case studies synthesis: a thematic, cross-case, and narrative synthesis worked example. *Empirical Software Engineering* 20, 6 (2015), 1634–1665.

[9] Bill Curtis, Herb Krasner, and Neil Iscoe. 1988. A field study of the software design process for large systems. *Commun. ACM* 31, 11 (1988), 1268–1287.

[10] Joanna F Defranco and Philip A Laplante. 2017. Review and analysis of software development team communication research. *IEEE Transactions on Professional Communication* 60, 2 (2017), 165–182.

[11] Tore Dyba, Barbara A Kitchenham, and Magne Jorgensen. 2005. Evidence-based software engineering for practitioners. *IEEE software* 22, 1 (2005), 58–65.

[12] Clyde W. Ford. 2021. *Think Black: A Memoir.* HarperCollins Publishers Inc.

[13] Nahum Gershon and Ward Page. 2001. What storytelling can do for information visualization. *Commun. ACM* 44, 8 (2001), 31–37.

[14] Amir Hossein Ghapanchi and Aybuke Aurum. 2011. Antecedents to IT personnel's intentions to leave: A systematic literature review. *Journal of Systems and Software* 84, 2 (2011), 238–249.

[15] Daniel Graziotin, Fabian Fagerholm, Xiaofeng Wang, and Pekka Abrahamsson. 2017. Consequences of unhappiness while developing software. In *2017 IEEE/ACM 2nd International Workshop on Emotion Awareness in Software Engineering (SEmotion).* IEEE, 42–47.

[16] John Grundy, Hourieh Khalajzadeh, and Jennifer Mcintosh. 2020. Towards Human-centric Model-driven Software Engineering.. In *ENASE.* 229–238.

[17] Jonathan Haidt. 2012. *The righteous mind: Why good people are divided by politics and religion.* Vintage.

[18] John Heron. 1996. *Co-operative inquiry: Research into the human condition.* Sage.

[19] Michael GeePaw Hill. 2021. *Tweet.* Retrieved February 18, 2021 from https://twitter.com/GeePawHill/status/1348697592851525635

[20] Michael GeePaw Hill. 2021. *Tweet.* Retrieved February 18, 2021 from https://twitter.com/GeePawHill/status/1348697719389483008

[21] Caitlin Kelleher and Randy Pausch. 2007. Using storytelling to motivate programming. *Commun. ACM* 50, 7 (2007), 58–64.

[22] Gene Kim. 2019. *The Unicorn Project: A Novel about Developers, Digital Disruption, and Thriving in the Age of Data.* IT Revolution Press.

[23] Gene Kim, Kevin Behr, and George Spafford. 2018. *The Phoenix Project: A Novel about IT, DevOps, and Helping Your Business Win.* IT Revolution Press.

[24] Per Lenberg, Robert Feldt, Lars Göran Wallgren Tengberg, Inga Tidefors, and Daniel Graziotin. 2017. Behavioral software engineering-guidelines for qualitative studies. *arXiv preprint arXiv:1712.08341* (2017).

[25] Per Lenberg, Robert Feldt, and Lars Göran Wallgren. 2015. Behavioral software engineering: A definition and systematic literature review. *Journal of Systems and software* 107 (2015), 15–37.

[26] Patrick J Lewis. 2011. Storytelling as research/research as storytelling. *Qualitative Inquiry* 17, 6 (2011), 505–510.

[27] Artur Lugmayr, Erkki Sutinen, Jarkko Suhonen, Carolina Islas Sedano, Helmut Hlavacs, and Calkin Suero Montero. 2017. Serious storytelling–a first definition and review. *Multimedia tools and applications* 76, 14 (2017), 15707–15733.

[28] Wayne G Lutters and Carolyn B Seaman. 2007. Revealing actual documentation usage in software maintenance through war stories. *Information and Software Technology* 49, 6 (2007), 576–587.

[29] Mithra Moezzi, Kathryn B Janda, and Sea Rotmann. 2017. Using stories, narratives, and storytelling in energy and climate change research. *Energy Research & Social Science* 31 (2017), 1–10.

[30] Emily Naul and Min Liu. 2020. Why story matters: A review of narrative in serious games. *Journal of Educational Computing Research* 58, 3 (2020), 687–707.

[31] Julian E Orr. 1986. Narratives at work: Story telling as cooperative diagnostic activity. In *Proceedings of the 1986 ACM conference on Computer-supported cooperative work.* 62–72.

[32] Jan-Peter Ostberg, Daniel Graziotin, Stefan Wagner, and Birgit Derntl. 2020. A methodology for psycho-biological assessment of stress in software engineering. *PeerJ Computer Science* 6 (2020), e286.

[33] David Lorge Parnas and Paul C Clements. 1986. A rational design process: How and why to fake it. *IEEE transactions on software engineering* 2 (1986), 251–257.

[34] Donald E Polkinghorne. 1995. Narrative configuration in qualitative analysis. *International journal of qualitative studies in education* 8, 1 (1995), 5–23.

[35] Austen Rainer. 2017. Using argumentation theory to analyse software practitioners' defeasible evidence, inference and belief. *Information and Software Technology* 87 (2017), 62–80.

[36] Paul Ralph, Nauman bin Ali, Sebastian Baltes, Domenico Bianculli, Jessica Diaz, Yvonne Dittrich, Neil Ernst, Michael Felderer, Robert Feldt, Antonio Filieri, Breno Bernard Nicolau de França, Carlo Alberto Furia, Greg Gay, Nicolas Gold, Daniel Graziotin, Pinjia He, Rashina Hoda, Natalia Juristo, Barbara Kitchenham, Valentina Lenarduzzi, Jorge Martínez, Jorge Melegati, Daniel Mendez, Tim Menzies, Jefferson Molleri, Dietmar Pfahl, Romain Robbes, Daniel Russo, Nyyti Saarimäki, Federica Sarro, Davide Taibi, Janet Siegmund, Diomidis Spinellis, Miroslaw Staron, Klaas Stol, Margaret-Anne Storey, Davide Taibi, Damian Tamburri, Marco Torchiano, Christoph Treude, Burak Turhan, Xiaofeng Wang, and Sira Vegas. 2021. Empirical Standards for Software Engineering Research. arXiv:2010.03525 [cs.SE]

[37] Kurt Schneider and J-P Von Hunnius. 2003. Effective experience repositories for software engineering. In *25th International Conference on Software Engineering, 2003. Proceedings.* IEEE, 534–539.

[38] Gerhard Schwabe, Alexander Richter, and Erik Wende. 2019. Special issue on storytelling and information systems. *Information Systems Journal* 29, 6 (2019), 1122–1125. https://doi.org/10.1111/isj.12258 arXiv:https://onlinelibrary.wiley.com/doi/pdf/10.1111/isj.12258

[39] Victoria A Shaffer, Elizabeth S Focella, Andrew Hathaway, Laura D Scherer, and Brian J Zikmund-Fisher. 2018. On the usefulness of narratives: an interdisciplinary review and theoretical model. *Annals of Behavioral Medicine* 52, 5 (2018), 429–442.

[40] Victoria A Shaffer and Brian J Zikmund-Fisher. 2013. All stories are not alike: a purpose-, content-, and valence-based taxonomy of patient narratives in decision aids. *Medical Decision Making* 33, 1 (2013), 4–13.

[41] Susan Elliott Sim and Thomas A Alspaugh. 2011. Getting the whole story: an experience report on analyzing data elicited using the war stories procedure. *Empirical Software Engineering* 16, 4 (2011), 460–486.

[42] Susan Elliott Sim, Thomas A Alspaugh, and Ban Al-Ani. 2008. Marginal notes on amethodical requirements engineering: what experts learned from experience. In *2008 16th IEEE International Requirements Engineering Conference.* IEEE, 105–114.

[43] Susan Elliott Sim, Janice Singer, and Margaret-Anne Storey. 2001. Beg, Borrow, or Steal: Using Multidisciplinary Approaches in Empirical Software Engineering Research. *Empirical Softw. Engg.* 6, 1 (March 2001), 85–93. https://doi.org/10.1023/A:1009809824225

[44] Janice Singer and Norman G. Vinson. 2002. Ethical issues in empirical studies of software engineering. *IEEE Transactions on Software Engineering* 28, 12 (2002), 1171–1180.

[45] Lemn Sissay. 2019. *My name is why.* Canongate Books.

[46] Joel Spolsky. 2006. *The Language Wars.* Retrieved February 18, 2021 from https://www.joelonsoftware.com/2006/09/01/language-wars/

[47] Will Storr. 2020. *The Science of Storytelling: Why Stories Make Us Human and how to Tell Them Better.* Abrams.

[48] StoryCollider. 2021. *The Story Collider.* Retrieved March 9, 2021 from https://www.storycollider.org/

[49] StoryCorps. 2021. *StoryCorps.* Retrieved March 9, 2021 from https://storycorps.org/

[50] E Tang, D Bleys, and N Vliegen. 2016. Making Sense of Adoptive Children's Inner World. Using Narrative Story Stem Techniques: A Systematic Review. *Manuscript in preparation* (2016).

[51] William Twining. 1994. *Rethinking evidence: Exploratory essays.* Northwestern University Press.

[52] Axel Van Lamsweerde. 2001. Goal-oriented requirements engineering: A guided tour. In *Proceedings fifth ieee international symposium on requirements engineering.* IEEE, 249–262.

[53] Karl E Weick. 1995. *Sensemaking in organizations.* Vol. 3. Sage.

[54] Anna Wiener. 2020. *Uncanny Valley: Seduction and Disillusionment in San Francisco's Startup Scene.* Fourth Estate.

[55] Emily Winter, Steve Forshaw, and Maria Angela Ferrario. 2018. Measuring human values in software engineering. In *Proceedings of the 12th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement.* 1–4.