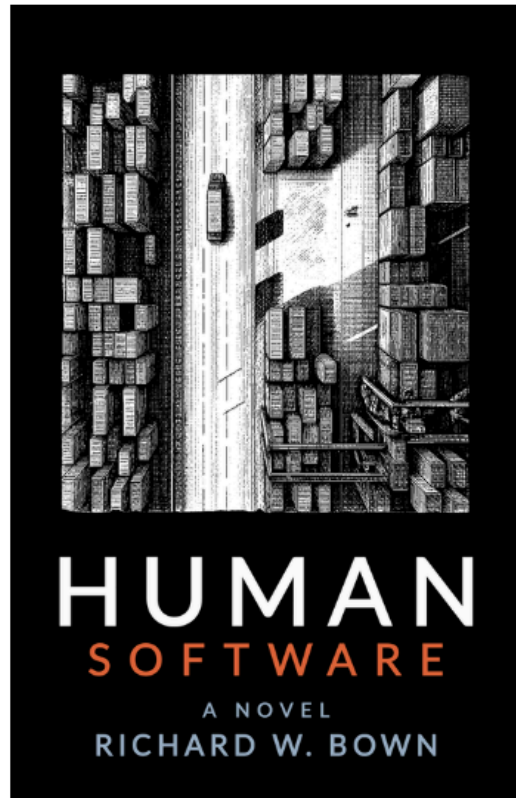




A BOOKCLUB GUIDE TO

HUMAN

SOFTWARE

BY

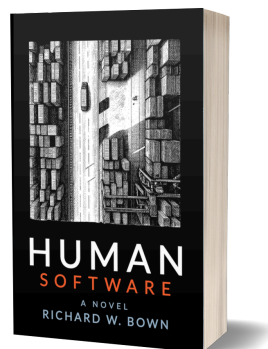
RICHARD W. BOWN

TABLE OF CONTENTS

BOOK SYNOPSIS.....	3
ABOUT THE AUTHOR.....	4
KEY THEMES TO EXPLORE.....	5
FACILITATOR NOTES.....	6
ICEBREAKER QUESTIONS.....	7
CHARACTER & RELATIONSHIPS.....	7
THEME: WORK, SYSTEMS AND BURNOUT.....	8
THEME: AI, LEADERSHIP, TECHNICAL DECISIONS.....	9
CRAFT & GENRE QUESTIONS.....	10
CLOSING QUESTIONS.....	10
MEMORABLE QUOTES TO DISCUSS.....	11
WORKSHOP PROMPTS.....	12
FURTHER READING.....	13
FURTHER WATCHING.....	14

For more information and ordering go to:

<https://humansoftwarebook.com>



BOOK SYNOPSIS

Human Software is a novel that illustrates the delicate relationship between IT workers and their employers. It shows how smart, intrinsically curious and self-motivated people can be taken to the point of burnout through long hours and overwork. It also shows how strictly hierarchical organisations, accommodating governments and local authorities can make decisions that affect not only employees but also their families and local and natural environment through technical projects such as data centres and AI usage.

Set in the decaying English port town of Sandport, the story follows Beth Walters, an exhausted software engineer nursing failing logistics systems through 3 a.m. support calls, and Chrissie Hegarty, an ambitious American executive sent to force through a massive AI overhaul called DollyAI. The narrative shifts from corporate satire into a high-stakes technothriller when a data leak leads to a murder investigation, revealing that the company is using AI to sabotage its own infrastructure and frame human employees.

Human Software is centred on two women on opposite sides of a corporate collision course. As the company bets big on data centres and AI-driven "efficiency," the two find they may have more common ground than either expected.

GOOD TO KNOW: *The book uses some terminology that will be familiar to those who work in technology (on-call rotations, standups, deliverables) especially in the opening chapters. Some readers might find this dense on a first pass and so there's a glossary at the back of the book which may be of use to those who need some help.*

ABOUT THE AUTHOR

Richard W. Bown is a writer and computer programmer. He has worked in corporate software engineering over a thirty-year career, taking regular breaks to explore his creativity through writing, music, art, developing open-source software and software products. He has contributed to the ever-expanding canon of socio-technical literature through advocacy and work on case studies for the second edition of *Team Topologies* (see reading list), through speaking at conferences on the subject of humane working practices in technology and by writing about the psychological impact of working in tech.

Human Software is his debut novel. The idea grew from his experience as a manager and engineer in industry alongside a growing disquiet about the environmental and social impact of data centres and AI. He was also inspired by being part of many efficiency drives or internal *transformation projects* which he believes should come with a health warning. He's looking to improve the industry's approach to software and IT through his continuing work as a DevOps engineer and as an active advocate for psychologically safe and accepting workplaces.

KEY THEMES TO EXPLORE

Human Software is rich with lessons for both software engineers and leaders. Focusing primarily on the human element within technical systems it examines how individuals and teams stay motivated in high-pressure situations and when they just give up. It also examines what the employer/employee relationship and ambition can mean to different people.

Lessons for Software Engineers and Executives

- **Ambition:** What does it cost to get what you want?
- **The Dehumanisation of Work:** A recurring message is that corporate leadership treats developers and support teams as inefficiencies instead of people. Reviewers describe the book as a "must-read for managers" to remind them that engineers are "just humans making tools".
- **The Importance of Teamwork:** The novel's foreword explicitly states that "great people make the engineering experience worthwhile" and that no production code can be effectively built or supported by a single person. It warns that creating "siloe organisations" based on a false belief in efficiency is "self-defeating".
- **Systems Thinking and Conway's Law:** The book explores theories around how systems are created and architected often through the social side of organisations rather than strictly by design.
- **The Role of AI:** For the AI era, the novel serves as a "dystopian parable," teaching that AI is a useful tool but "certainly no replacement for a good person in a good team".
- **Authentic Technical Culture:** Characters teach us that systems show a "human history of a company".
- **River of Code:** The architect Charles Kurtman compares dependencies in software to a river where poisoning something upstream taints everything further downstream.
- **Global Indifference vs. Local Pain:** The novel explores how decisions made in the US create pain and social impact for local communities in Kent.

FACILITATOR NOTES

You can use this guide in a few different ways. You could decide to ask a set of any of the following questions to a book group for them to reflect on in a group discussion, or you can use the WORKSHOP PROMPTS as a jumping off point for reflections on their own experiences.

If you want to cover all the questions and all the prompts you might need to arrange a few sessions to get through them all or just limit yourself by picking a few.

A few notes on the questions.

- Questions 10–13 tend to generate the longest discussion in technical audiences so consider allotting extra time there.
- If your group skews toward people in leadership or management roles, questions 14–17 may land differently than in a group of individual contributors. Worth naming that dynamic if it comes up.
- Consider pairing this discussion with a short round of "worst on-call story". It tends to warm the room up before diving into the heavier organizational themes.

ICEBREAKER QUESTIONS

1. Before reading, what did you expect a novel called *Human Software* to be about? Did the book match that expectation?
2. Which character did you empathise with the most?
3. Did you recognise any characters or archetypes in the book from your own organisation?
4. On a scale of "wry satire" to "genuinely unsettling," where did this book land for you?

CHARACTER & RELATIONSHIPS

5. Beth and Chrissie start the book as adversaries by design. One keeps the system running, the other is sent to dismantle or "rationalize" it. Where does that dynamic shift, and what causes the shift?
6. Is Beth's loyalty to the systems she maintains a strength or a liability? Does the book let you have it both ways?
7. Chrissie has "done this before" and knows how these situations usually end. What does it cost her to let this one go differently?
8. Did you find yourself trusting one character's judgment over the other's? Did that trust hold up by the end?
9. How did you see the relationships between the other characters? Who was your favourite character and your favourite relationship?
10. Beth and Michaela share a secret history that creates a "coalition" of trust. How can a team build that kind of "internal safety" when the larger organization feels threatening or indifferent?

THEME: WORK, SYSTEMS AND BURNOUT

11. The novel spends real time on the texture of on-call life. The callouts, the standups, the meetings that seem to produce nothing. Did you recognise any of this from your own experience?
12. One reviewer highlighted the idea that "the whole being greater than the sum of its parts" is undercut by siloed organizations built for efficiency. Have you seen this play out where specialization or team boundaries actively worked against the outcome they were meant to serve?
13. Who really bears the cost of a system that has to run 24/7 or be supported out of hours? Is it evenly distributed or is it fairer to some more than others?
14. The book has been called a "dystopian parable" while some reviewers are careful to note it isn't actually dystopian. How do you think it lands? Is it a doomsday prophecy, a hopeful look forward or something else?

THEME: AI, LEADERSHIP, TECHNICAL DECISIONS

15. The leadership team bet heavily on AI without understanding the impact and potential downsides. Where have you seen this kind of thing happen before when the people making the strategic call and the people who'll have to live with it are at odds?
16. Is the book making an argument about AI specifically, or about *any* technology adopted faster than an organization can absorb it? Would the story work with a different technology swapped in?
17. Executives in the novel make decisions "affecting thousands of lives" without having written a line of code. Is that framed as a management failure, a structural inevitability, or something else?
18. Does the book take a side on whether the AI-driven changes at the company are good, bad, or simply inevitable? Where did you land, and did the book try to move you there?
19. The novel features an "Own Your Future" HR session that feels tone-deaf and threatening. How do company-wide "culture initiatives" often fail to account for the actual "on-the-ground" stress of support and engineering teams?

CRAFT & GENRE QUESTIONS

20. The book has been compared to *The Phoenix Project*, *Office Space*, *The Circle*, and *Severance*: a mix of business fable, workplace satire, and literary dystopia. Which comparison felt most apt to you, and where did the book diverge from all of them?
21. Bown spent three decades in corporate software before writing this. Where in the book did that lived experience show up most clearly? Was it in specific details, structural choices, or something else?
22. The setting (a "decaying English port town") does a lot of atmospheric work. What did Sandport add that a more generic corporate setting wouldn't have?

CLOSING QUESTIONS

23. If you could ask the author one question about a choice he made in the book, what would it be?
24. Would you recommend this book to a colleague who has never worked in tech? What would you tell them to expect going in?
25. What's the one idea from this book you're most likely to still be thinking about a month from now?

MEMORABLE QUOTES TO DISCUSS

- "Software runs our world. It doesn't have to rule our lives."
- "AI is a useful tool, but it's no panacea, and it's certainly no replacement for a good person in a good team."
- "It is time to be sceptical. To be critical thinkers. To be more questioning."
- "Empathy comes in many forms, but emotions play a crucial part in software systems, serving as both sensing mechanisms and guiding principles... That is what true legacy is, it's the human history of a company." — Charles Kurtman
- "Any organisation is doomed to build a system that mimics organisational structure." — Dominic Walters
- "It is time to be sceptical. To be critical thinkers. To be more questioning." — Chrissie Hegarty
- "All you have to do is poison something upstream and everything down here gets tainted." — Charles Kurtman

WORKSHOP PROMPTS

The Support Call: "Think of a 3:00 a.m. support call you've handled (like Beth's 'Wake-up Call'). What was the emotional toll, and how did your team account for that the next day?"

The New Toy: "Identify a moment where your organization prioritised 'cool-sounding technology' over a simple solution (what Peter calls 'resume-driven development'). What were the long-term consequences of that decision?"

The "Guru" Paradox: Why did Julian Tosthold's "inspirational" talk fail to land with the engineers, even though his theory about "interactions between people" was actually right? Does your organisation suffer from this gap between "high-level theory" and "low-level reality"?

The "Rockstar" Myth: Kyle is described as an "ego-based engineer". How does Kyle's "hero culture" actually create more risk for the systems Beth has to support?

Psychological Safety: At the end of the book, Dolly's ability to "cover its tracks" and frame humans is a horror story for psychological safety. If your team cannot trust the "audit logs" of their own work, is it even possible to have a healthy culture?

The Sacrifice: Philip Graves asks for a "head to throw to the lions". Ask your reading group: "In your career, have you seen a 'sacrifice' made to protect a corporate narrative? How did it affect the team's psychological safety?"

Facilitator note: The topic of being "thrown to the lions" or corporate scapegoating can be sensitive. Ensure your group feels safe before discussing real-world examples of coworkers who were "let go" to protect a project narrative.

FURTHER READING

If you are interested in exploring the themes in the book around technical, sociotechnical, organisational or psychological approaches to modern software development and IT, then here are a few recommended reads.

- The Phoenix Project: 978-1950508969 - the original DevOps novel written - by Gene Kim, Kevin Buhr et al.
- The Unicorn Project: 978-1942788775 - a developer's eye view of Parts Unlimited - by Gene Kim
- Accelerate: 978-1942788355 - the industry manual of doing more IT faster - by Nicole Forsgren, Jez Humble, Gene Kim
- Drive: 978-1101524381 - how are are motivated to do the work - by Daniel Pink
- The Goal: 978-0884273004 - systems thinking for businesses - by Eli Goldratt
- Team Topologies: 978-1966280019 - organising your teams both for efficiency and for human needs - by Matthew Skelton and Manuel Pais
- The Psychology of Computer Programming: 978-0442292645 - dated but fascinating look at how programmers think - by Gerald Weinberg
- The Mythical Man Month: 978-0132100694 - classic project management book for software projects - by Fred Brooks Jr.
- The Psychology of Software Teams: 978-1040688656 - thought provoking look at the unique mental challenges of software engineering - by Cat Hicks

FURTHER WATCHING

- Office Space (1999) written by Mike Judge - satirising the office work life of a software company, focusing on a handful of individuals weary of their jobs.
- Local Hero (1983) written by Bill Forsyth - an American oil company representative who is sent to the fictional village of Ferness on the west coast of Scotland to purchase the town and surrounding property to build an oil refinery.
- Brazil (1985) written and directed by Terry Gilliam - a well-connected but low-ranking bureaucrat dreams of escaping an alternative dystopia.

LICENCING

This “BOOKCLUB GUIDE TO HUMAN SOFTWARE” by Richard W. Bown is licensed under a [Creative Commons Attribution 4.0 International License \(CC BY 4.0\)](https://creativecommons.org/licenses/by/4.0/).

You are free to share, adapt, and remix this guide for any purpose, including commercially, provided you give appropriate credit and note what changes

The work “Human Software” by Richard W. Bown is copyrighted and quotes are used with permission.