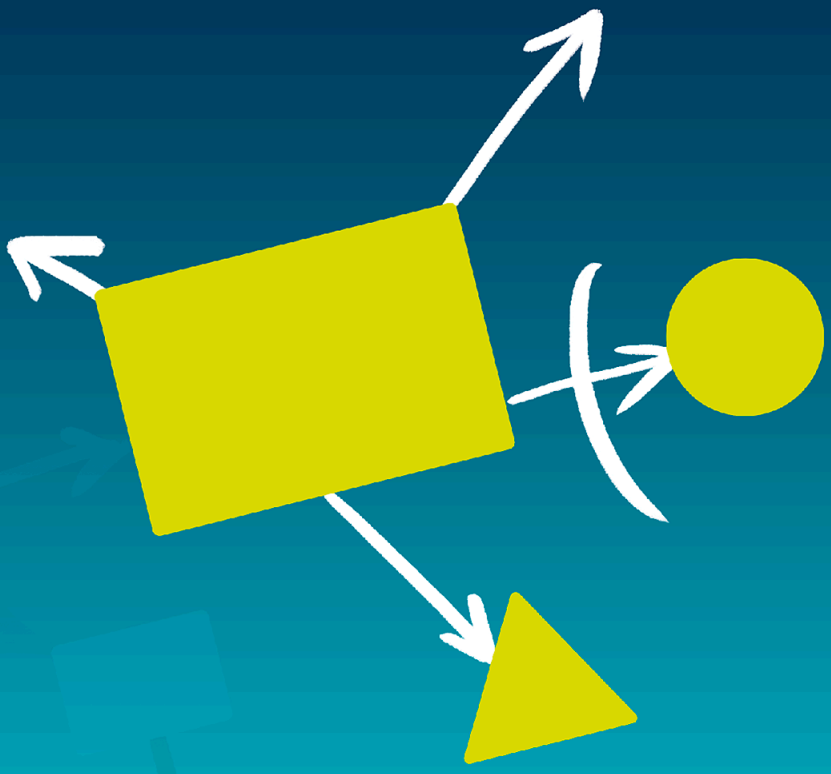
BETTER WHITEBOARD SKETCHES

How to Sketch Clear Technical Diagrams

MATTHEW SKELTON



conflux

Better Whiteboard Sketches

How To Sketch Clear Technical Diagrams

Matthew Skelton

This book is for sale at
http://leanpub.com/BetterWhiteboardSketches

This version was published on 2023-02-25

ISBN 978-1-912058-95-2

conflux

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Contents

1. How to use this book

This book provides a very practical approach to better whiteboard sketches. Although the techniques are based on sound research and practical usage, the theory is kept to a minimum and you can get started with techniques almost immediately.

Here are some suggestions for using the book:

- To **remind yourself of the techniques**, print or see the *Quick Reference* guide.
- To **practise the techniques**, use the exercises at the end of each chapter and see the Case Studies.
- To understand how to use whiteboard sketches for **better conversations** and **in-erson explanations**, see Chapter 3
- To learn how to **use colour effectively** in whiteboard sketches, see Chapter 2
- To learn how and why to **identify and represent key concepts**, see Chapter 1

The three central chapters each contain a set of exercises for you to try out the techniques you have learnt.

1.1 Conventions used in the book

A box with a whiteboard symbol shows an important aspect of *Better Whiteboard Sketches*:

Decide on the *purpose* of the sketch before starting.

A box with a key symbol shows a useful tip:

To remember things better, sketch more.

A callout with an 'i' symbol shows some general information:

Although the examples used in this book are from the field of software systems, the techniques are applicable to a wide range of technical disciplines.

A pen writing indicates exercises for the reader.

1. Draw a square with thicker edges on the sides

2. Introduction - the value of clear diagrams

Key points

- Aim to make whiteboard sketches clear and valuable for a long time
- Sketching helps us to understand and remember
- Take advantage of embodied cognition by using a physical whiteboard

This book provides a simple, proven approach to drawing clear, memorable technical diagrams. Although the focus is on whiteboard sketches, the principles apply to chalkboards, paper notebooks, and even digital tools. The techniques are applicable to a wide range of technical disciplines, including:

- software architecture
- systems architecture
- business process modelling
- chemical engineering

- mechanical engineering and robotics
- electronic and electrical engineering

The approach is designed to be suitable for people with a range of learning and visual needs, including:

- people with dyslexia and other learning differences
- people with visual impairments in colour or acuity
- people who learn visually or audibly
- people who like to categorise in order to understand

The techniques lead to diagrams that still make sense many months or years after drawing them. The diagrams tend to be clear enough to include directly in slide decks and even official documentation. The techniques make it simple and quick to redraw diagrams if needed, without having to begin from the very start each time.

A well-drawn diagram can help a diverse group of people to understand and remember concepts and details, saving time and effort in avoiding repetition and misunderstandings. People who draw clear diagrams gain a reputation for being more understandable and helpful and their ideas and approaches are therefore more likely to succeed.

Recent research has begun to acknowledge that "reliance on [digital] information management infrastructures is probably not sufficient to enable organizations to utilize their intellectual capital effectively" [8]. Digital tools are hugely powerful and valuable, but analogue, physical tools and techniques - such a whiteboard sketching - also have an important place, as they help to "facilitate interaction and turn-taking, increase vividness and memorability".

Through the companion training workshops, the techniques in this book have been tested and refined with people from around the world. “A picture is worth a thousand words” goes the English saying, but too often whiteboard sketches and notebook drawings quickly lose their meaning after they are drawn. If you have ever looked at a technical diagram and thought “these squiggles are meaningless!”, then this book is for you.

Remember: although the examples used in this book are from the field of software systems, the techniques are applicable to a wide range of technical disciplines.

2.1 Why sketch?

Both artists and architects of buildings have long valued sketches as a key part of the design process [2]. In particular, the sketches by Leonardo da Vinci demonstrate how value sketching was to this great artist and designer [7]; although the technical sketches we make to design systems and products are rather more humble than those of da Vinci, we should expect to refer back to them as our designs progress.

Award-winning software architect Ruth Malan has studied the notebooks of Renaissance artist and inventor Leonardo da Vinci renowned for his sketches 10 and identified a wide range of reasons for sketching:

- To observe (more attentively)
- To study, think, reason

- To record
- To invent, to combine, to make connections
- To test ideas
- To persuade

Some people have a negative view of whiteboard sketching, that it's somehow not "real work", but this is misguided. When we use whiteboard sketches to test ideas, invent, study, observe, make new connections, and persuade people, sketching is clearly valuable work. In particular, whiteboard sketching can avoid hours, day, weeks, or months of *re-work* to correct an implementation that was based on faulty assumptions.

As we will see in Chapter 1, before starting a sketch it is valuable to take a moment to consider *why* we're drawing a whiteboard sketch. A sketch to help us study or observe might include very different elements to a sketch whose primary purpose is to persuade other people.

Decide on the *purpose* of the sketch before you start drawing.

2.2 Typical problems with whiteboard sketches

Whiteboards are great for rapid sketching of concepts, ideas, interrelationships, data flows, systems, and many other technical purposes. The transient nature of the drawings helps people to focus on communicating ideas and details because changes are quick and easy. However, this ease of use comes

at a price: many whiteboard sketches are difficult to understand, leading to confusion and wasted time.

In his excellent article *The Art of Crafting Architectural Diagrams* [9], Ionut Balosin lists many of the typical problems with whiteboard sketches, including:

- Missing key or legend - what do different shapes mean?
- Ambiguous arrows - do arrows represent ownership, data flow, or something else?
- Ambiguous colours - do the colours have a consistent meaning?
- Confused concepts - what level of detail/abstraction is being showm?

The advice and approaches in this book address problems like these through the use of simple, structured rules and heuristics for clearer whiteboard sketches.

2.3 Drawing helps us to understand and remember

Another reason to use sketches as a regular part of the design process is because sketching actually improves memory formation and recall. In a study published in 2016, Jeffrey Wammes and colleages from the Universtiy of Waterloo in Ontario, Canada conducted a series of experiments to assess the effect on drawing on memory and recall of information [1]. They concluded that:

> "drawing improves memory by encouraging a seamless integration of semantic, visual, and motor aspects of a memory trace"

So drawing sketches helps us to remember things better than simply writing things down, which suggests that if we want to remember things better, we should sketch more. {icon=chalkboard} G> To remember things better, sketch more.

2.4 Use a physical whiteboard

Over the past decade or so there has been increasing awareness that their bodies strongly influence the way that humans learn and think, a concept called *embodied cognition* [11]. With embodied cognition, "knowledge is grounded in sensorimotor systems, and that learning can be facilitated to the extent that lessons can be mapped to these systems" [12].

In the context of sketches, there is significant value to be gained from interacting with a physical whiteboard rather than a software version at a single computer. Large digital whiteboards can be very effective *if* they use digital pens, not a standard mouse or trackpad pointer.

In a study published in 2016, researchers used a variety of learning environments (including whiteboards) to assess the effect of physical interaction with tools on retention of learned information [12]. They found that:

> better retention of certain types of knowledge can be seen over time when more embodiment is present during the encoding phase

Information seems to be more strongly "encoded" in the brain when the person learning is using their body in a physical way. People who can use physical whiteboards should take advantage of this: prefer large physical whiteboards - and the

act of physical drawing with pens on a large surface - and enjoy the increased learning and retention that comes with embodied cognition!

Although wall-mounted whiteboards are probalby the most effective type, even small/portable whiteboards can be effective for diagramming. Portable whiteboards have the advantage that you can easily take to whiteboard to another part of the office to show a colleague or move the whiteboard to a room with a video conferencing camera for a video call.

Some digital whiteboard technologies encourage embodied cognition due to their size and interaction tools. For example, the Microsoft Surface Hub has a Whiteboard feature that is quite natural using physical-digital pens on the screen. In this book, a digital whiteboard with natural physical interaction counts as a physical whiteboard.

Use a physical whiteboard (or chalkboard) for enhanced learning - or a digital tool that feels like a physical whiteboard.

2.5 Take photographs and share

After a useful whiteboard discussion, take photos of the whiteboard sketch and upload to a wiki or group chat to share with colleagues. If your workplace forbids photography, try to use a digital whiteboard from which you can save an image directly.

There are several free software applications that automatically enhance photos of whiteboards, including Office Lens and Scanbot. Useful features include deskewing (so the image is straightened), colour correction, and auto-brightening.

2.6 3 core principles: concepts, colour, repeating

This book sets out three core principles for drawing better whiteboard sketches, which are covered in the nex three chapters:

1. Define and group key concepts - Chapter 1
2. Use colour but think in monochrome - Chapter 2
3. Use multi-stage sketches to help discussions - Chapter 3

Any one of these principles will improve a whiteboard sketch, but when used together they form a powerful approach to more understandable, more useful diagrams.

2.7 Summary

Many whiteboard sketches are unintelligible to people who did not see the diagram being drawn. This leads to confusion and sometimes significant wasted effort for teams building and running systems or engaged in knowledge work.

Through embodied cognition, a physical whiteboard helps to encode learning better than software tools alone, and because whiteboard sketches are very useful for learning and exploring, using a physical whiteboard is best.

By considering the needs of people with different learning and perception needs, the *Better Whiteboard Sketches* approach produces clearer, more memorable sketches for everyone.

3. Define and group key concepts

(This chapter is not available in this edition of the book.)

4. Use colour but think in monochrome

(This chapter is not available in this edition of the book.)

5. Use multi-stage sketches for clarity

(This chapter is not available in this edition of the book.)

6. Case Studies and Quick Reference

In this chapter are some case studies from real-world situations to help illustrate the value of the techniques in the rest of the book. There is also a Quick Reference guide.

6.1 Case Studies

1 - Mobile-driven cloud application

Consider a mobile-driven cloud-based software application - this could be something like an appointment booking system, an insurance quotation system, or perhaps a way to manage a stocks & shares portfolio. [1] The mobile application sends requests via the internet to a cloud-based central system which (after traffic filtering) routes the requests via some orchestration services to a core API. The API might generate additional requests to a external supplier and/or to a legacy system hosted "on-premise" (in a data centre).

Here is an initial sketch of the system before the Better Whiteboard Sketches techniques were applied (Figure 4.1):

[1] Thanks to Hibri Marzook for permission to use these sketches and to BCS SPA Conference for hosting the workshop during which these sketches were made.

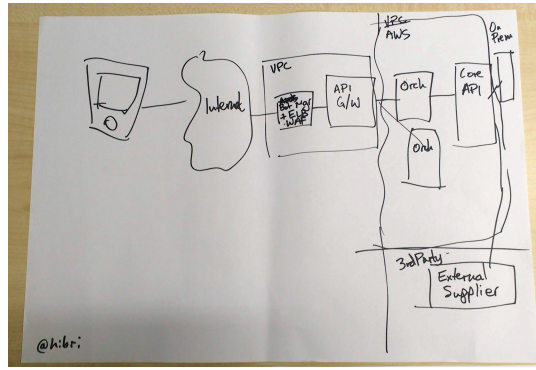


Figure 4.1 - Case Study 1 - before improvements

It is difficult to see immediately what some of the lines represent; the nature of the different boxes is also not very clear. Now let's see how the sketch was improved through the Better Whiteboard Sketches techniques (Figure 4.2):

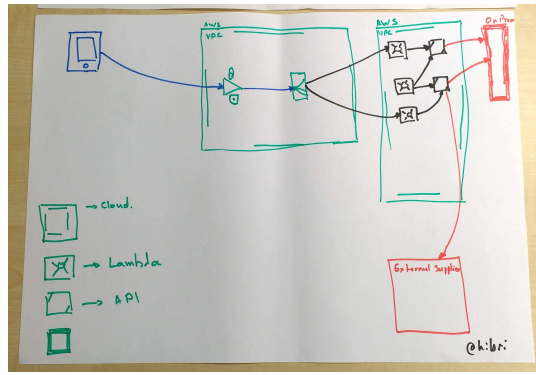


Figure 4.2 - Case Study 1 - after improvements: a key/legent, colours, different shapes, and line styles help to clarify the concepts

2 - Retail system with POS terminals

The software system for a large retailer includes a main sales application hosted in the cloud and also a smaller application

deployed locally to Point-of-Sale (POS) systems in each retail store. Stock control and email services are provided by a separate supplier. [2]

Figure 4.3 shows the initial sketch of the system before the Better Whiteboard Sketches techniques were applied. There is a nice use of broken lines to show some separation of different parts of the software, but other concepts are less easily distinguished:

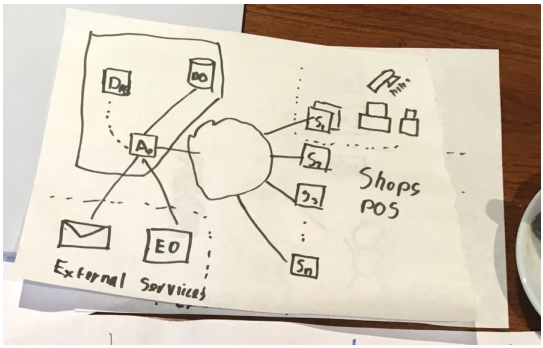


Figure 4.3 - Case Study 2 - before improvements

After applying some techniques from Better Whiteboard Sketches, the resulting diagram was much clearer (Figure 4.4):

[2]Thanks to Jildert Miedema for permission to use these sketches and to DDD EU conference for hosting the workshop during which these sketches were made.

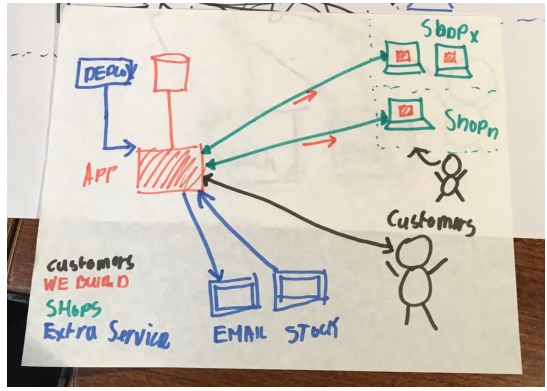


Figure 4.4 - Case Study 2 - after improvements: colour, hatching, shapes, and key/legend all clarify the diagram

Hatching, double lines, colours, and a key/legend make the sketch much more understandable and memorable.

3 - Sketches for Team Topologies book

> Matthew writes: "During 2018 and 2019 I co-authored the book Team Topologies with Manuel Pais [3] which looks at patterns for organisation dynamics and team interactions for modern software delivery. We wanted to ensure the diagrams were as legible and understandable as possible; even though the book would be printed in full colour, we wanted the diagrams to work well in monochrome [for the reasons given in Chapter 2]. I decided to use the techniques from the Better Whiteboard Sketches book to inform our approach to diagrams for the Team Topologies book."

In the Team Topologies book, we needed to represent four core types of team typically needed for modern software

[3] See teamtopologies.com/book for details of the Team Topologies book.

delivery (Stream-aligned, Enabling, Complicated Subsystem, and Platform). How should these different kinds of team be shown in diagrams? Figure 4.5 shows an early version of a whiteboard sketch showing thes four team types:

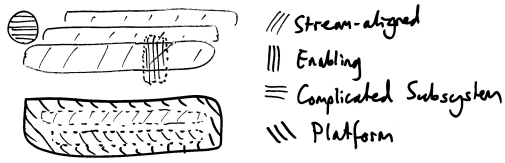


Figure 4.5 - Case Study 3 - initial team types sketch

Initially, the four types of team were represented only with different types of hatching/shading and some differentiation in shapes. This was a reasonable first version, but the hatching all looked rather similar. We applied various techniques from Better Whiteboard Sketches (colour, different hatching, different shapes, and double lines) to better distinguish the different concepts. The whiteboard version evolved to the diagram in Figure 4.6:

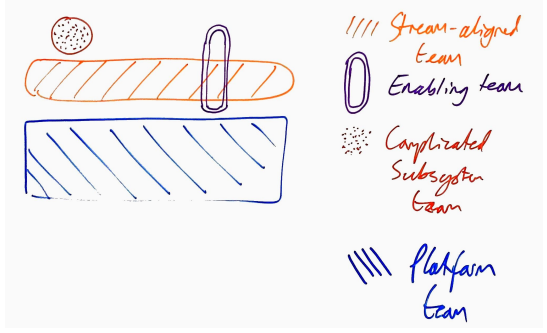


Figure 4.6 - Case Study 3 - improved team types sketch

In the final printed versions of the diagrams, we have different shapes for the different types of team (octagon, vertical and

horizontal rectangles), symbols and hatching for the team interactions, and colours that are not usually confused by people with colour vision deficiency (see Figure 4.7).

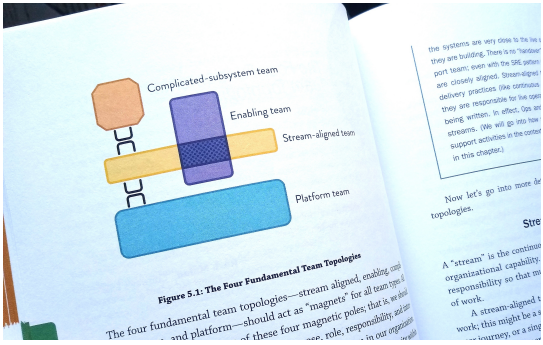


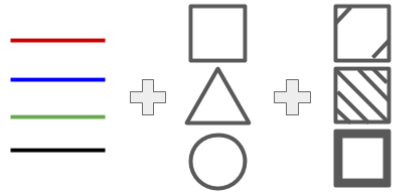
Figure 4.7 - Case Study 3 - final diagram in the Team Topologies book

The early whiteboard sketches helped in discussions with the graphic designer and editors of the book because through the iteration of the whiteboard sketches concepts became clearer and the designer was better able to create print-ready versions.

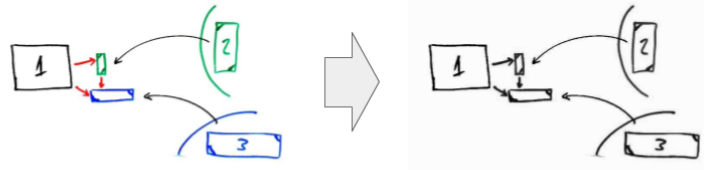
6.2 Quick Reference

1. Define and group key concepts

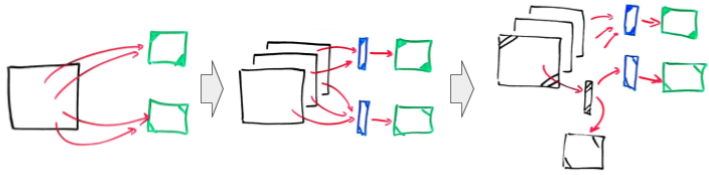
- Data Flow
- Ingress / Egress
- Ours vs Theirs
- VMs vs Functions
- Tests vs Products



2. Use colour but think in monochrome



3. Use multi-stage sketches to help discussions



Print this page as a useful guide. See BetterWhiteboardSketches.com for more tips and examples.

7. Terminology

- API: Application Programming Interface - a way to interact with software from other software (rather than via a screen)
- C4 Model: an approach to software architecture diagrams devised by Simon Brown - see c4model.com
- CDN: Content Delivery Network
- CVD: colour vision deficiency
- DDD: Domain Driven Design, a technique for modelling and understanding complicated business domains based on the book by Eric Evans.
- IoT: Internet of Things
- Key/Legend: a guide to the meaning of elements in a diagram
- LAN: Local Area Network
- monochrome: a single colour
- SaaS: Software as a Service
- UML: Unified Modelling Language, a diagramming scheme popular in the 1990s and 2000s that uses predefined diagram formats for various purposes in software engineering
- UX: User Experience
- VLAN: virtualized LAN

8. References

Due to limitations of the LaTeX text processor used by Leanpub to generate the book from the manuscript files, some of the URLs in this section may not display correctly. A full list of references is available on the book website.

Visit BetterWhiteboardSketches.com for details.

1) Wammes, Jeffrey D., Melissa E. Meade, and Myra A. Fernandes. "The Drawing Effect: Evidence for Reliable and Robust Memory Benefits in Free Recall." The Quarterly Journal of Experimental Psychology 69, no. 9 (September 1, 2016): 1752–76. https://doi.org/10.1080/17470218.2015.1094494

2) Arnheim, Rudolf. "Sketching and the Psychology of Design." Design Issues 9, no. 2 (1993): 15–19. https://doi.org/10.2307/1511669

3) Deleniv, S. "The Mystery of Tetrachromacy: If 12% of Women Have Four Cone Types in Their Eyes, Why Do so Few of Them Actually See More Colours?" The Neurosphere (blog), December 17, 2015. https://theneurosphere.com/2015/12/17/the-mystery-of-tetrachromacy-if-12-of-women-have-four-cone-types-in-their-eyes-why-do-so-few-of-them-actually-see-more-colours/

4) "Improving The Color Accessibility For Color-Blind Users." Smashing Magazine. https://www.smashingmagazine.com/2016/06/improvin color-accessibility-for-color-blind-users/

5) Ren, Fei. "What My Color-Blindness Taught Me about De-

sign." UX Collective, January 23, 2018. https://uxdesign.cc/what-my-color-blindness-taught-me-about-design-d3009a93ff9c

6) "Black and White Photography Tips: The 5 Cornerstones of All Great Monochrome Photos." GuruShots.com. Accessed June 25, 2018. https://gurushots.com/article/black-and-white-photography-guide

7) Victoria and Albert Museum. "V&A · Leonardo Da Vinci's Notebooks." Victoria and Albert Museum, July 26, 2016. https://www.vam.ac.u da-vincis-notebooks

8) Pfister, Roland A., and Martin J. Eppler. "The Benefits of Sketching for Knowledge Management." Journal of Knowledge Management 16, no. 2 (March 30, 2012): 372–82. https://doi.org/10.1108/1

9) Balosin, Ionut. "The Art of Crafting Architectural Diagrams." InfoQ (blog), August 4, 2017. https://www.infoq.com/articles/crafting architectural-diagrams

10) Malan, Ruth. "Design Visualization: Smoke and Mirrors (Slides)." Software, December 4, 2015. https://www.slideshare.net/RufM/desig visualization-smoke-and-mirrors-slides-55822413

11) Skulmowski, Alexander, and Günter Daniel Rey. "Embodied Learning: Introducing a Taxonomy Based on Bodily Engagement and Task Integration." Cognitive Research: Principles and Implications 3 (March 7, 2018). https://doi.org/10.1186/s41235-018-0092-9.

12) Johnson-Glenberg, Mina C., Colleen Megowan-Romanowicz, David A. Birchfield, and Caroline Savio-Ramos. "Effects of Embodied Learning and Digital Platform on the Retention of Physics Content: Centripetal Force." Frontiers in Psychology 7 (2016). https://doi.org/10.3389/fpsyg.2016.01819

13) Tune, Nick. "'Intentional Naivety First' Bounded Context Modelling." Medium (blog), February 8, 2018. https://medium.com/nick-

tune-tech-strategy-blog/intentional-naivety-first-bounded-context-modelling-62e6211574ec

14) RNIB. ‘Colour Vision Deficiency’. RNIB - See differently, 14 November 2017. https://www.rnib.org.uk/nb-online/colour-vision-deficiency

9. Acknowledgements

I'd like to thank several people for inspiration when writing this book: Ruth Malan for her visionary approach to software architecture and visualisation; Nick Rozanski and other people from SPA Conference for some very useful early feedback on the material; Simon Brown for his work on the C4 Model; all the people I have pair-sketched with over the years, especially Owain Perry, Rob Thatcher, and Manuel Pais; and all attendees of my *Better Whiteboard Sketches* workshops for their feedback and enthusiasm.

Matthew Skelton, February 2019

9.1 Licenses

This book uses icons from Font Awesome, used under a Creative Commons license https://fontawesome.com/license.

10. About the author

10.1 Matthew Skelton

Matthew Skelton

Matthew Skelton has been building, deploying, and operating commercial software systems since 1998. Head of Consulting at Conflux (confluxdigital.net), he specialises in Continuous Delivery, operability and organisation design for software in manufacturing, ecommerce, and online services, including cloud, IoT, and embedded software.

In 2015, Matthew co-founded pioneering DevOps consultancy Skelton Thatcher Consulting, which led industry thinking around operability and team design. Matthew is the co-author of several books including: Team Topologies: organizing business and technology teams for fast flow (IT Revolution, 2019), Internal Tech Conferences (Conflux Books, 2019), Continuous Delivery with Windows and .NET (O'Reilly, 2016), and Build Quality In (Conflux Books, 2015). Matthew also instigated the

first conference in Europe dedicated to Continuous Delivery practices for modern software, PIPELINE Conference.

In addition to holding a BSc degree from the Univeristy of Reading in Computer Science and Cybernetics, Matthew holds an MSc in Neuroscience from Oxford University, where he studied the effects of dyslexia on speech perception, and an MA in Music from the Open University, where he studied the accesibility of research materials to visually impaired musicians. He is a Chartered Engineer (CEng).

10.2 Why I wrote this book

> Over the course of consulting engagements around the world, I have drawn hundreds (maybe thousands!) of technical diagrams. A few years ago, I noticed that clients would comment on how effective and clear the diagrams were; as these comments became more frequent, I decided to "reverse engineer" my drawing techniques to try to understand what it was that people found so useful in the diagrams.
>
> The result of that reverse engineering is this book and the companion workshop training course *Better Whiteboard Sketches* that I have run at several public and private training sessions since 2018.
>
> I hope you find the book useful; I certainly have enjoyed distilling these techniques into this book and the companion workshop.

– *Matthew Skelton*

11. Conflux Books

Books for technologists by technologists

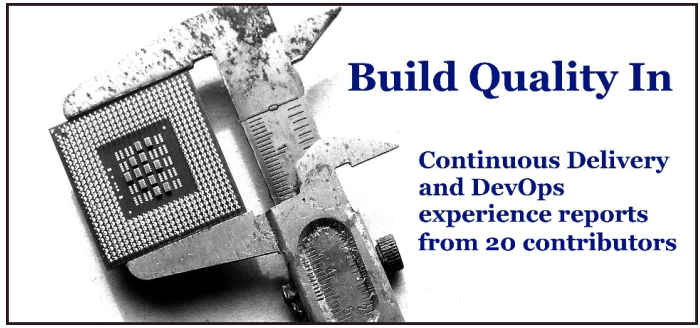
Our books help to accelerate and deepen your learning in the field of software systems. We focus on subjects that don't go out of date: fundamental software principles & practices, team interactions, and technology-independent skills.

Current and planned titles in the *Conflux Books* series include:

1. *Build Quality In* edited by Steve Smith and Matthew Skelton (B01)
2. *Better Whiteboard Sketches* by Matthew Skelton (B02)
3. *Internal Tech Conferences* by Victoria Morgan-Smith and Matthew Skelton (B03)
4. *Technical Writing for Blogs and Articles* by Matthew Skelton (B04)

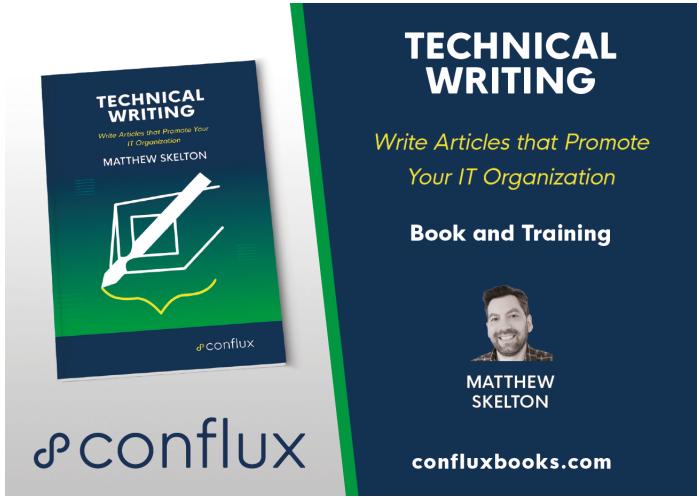
Find out more about the *Conflux Books* series by visiting: confluxbooks.com

conflux



Build Quality In - a book of Continuous Delivery and DevOps experience reports. Edited by Steve Smith and Matthew Skelton. Conflux Books, April 2015

Internal Tech Conferences by Victoria Morgan-Smith and Matthew Skelton. Conflux Books, April 2019



Technical Writing for IT by Matthew Skelton. Conflux Books, August 2019

Team Guides for Software: *Software Operability*, *Metrics for Business Decisions*, *Software Testability*, *Software Releasability*. Skelton Thatcher Publications, 2016-2019