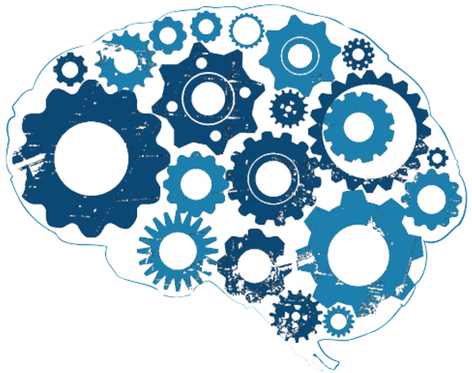
DEVOPSDAYS DETROIT

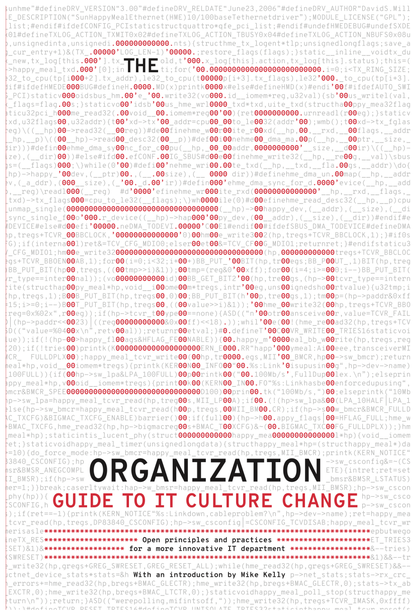
DEVOPS README.MD

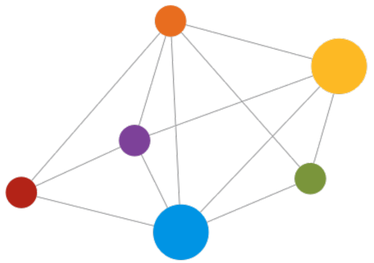
chrisshort.net

open source .com

DevOps'ish



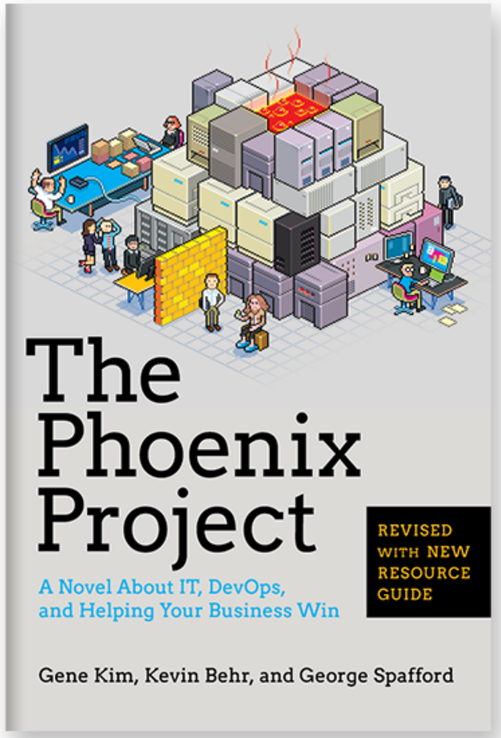




DZone

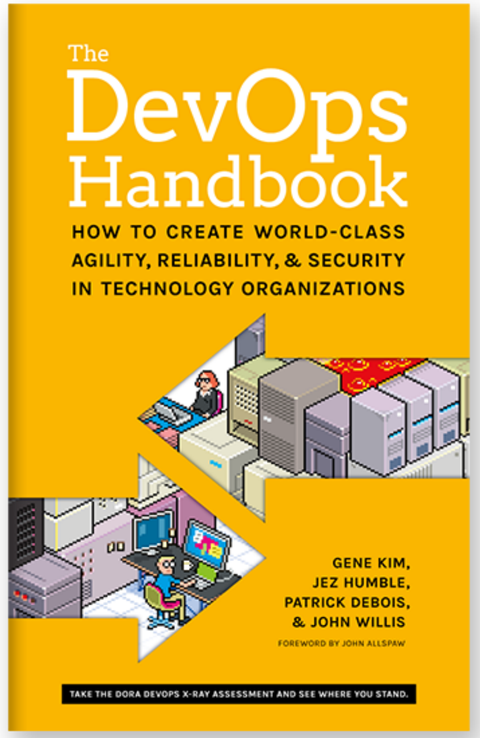
THE PHOENIX PROJECT

- Novel; Not your typical technical book
- Transformation of Broken Organization towards DevOps Culture
- Quintessential beginning of a DevOps journey
- Pros: Easy to digest, can suggest to executives
- Cons: The implementation details are fuzzy
- Quip: We all know Brent. Help Brent not be Brent.



THE DEVOPS HANDBOOK

- Handbook full of use cases and helpful examples
- Years of experience poured into one book
- The next step of a DevOps journey
- Pros: Detail oriented, can give to technical staff
- Cons: Not a quick read
- Quip: You're DevOps'ing if you quote this book.



THE TWELVE-FACTOR APP

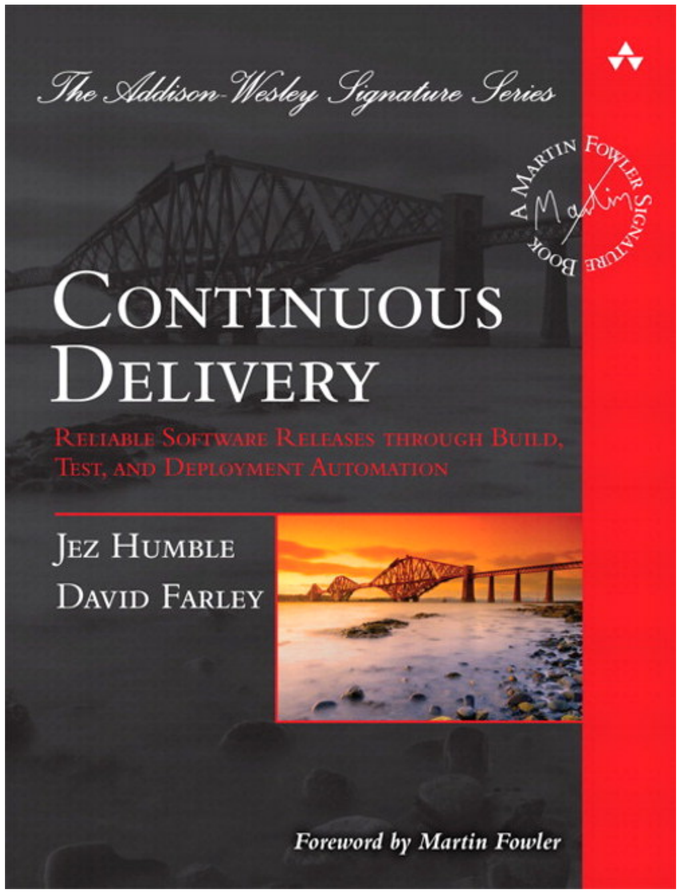
- https://12factor.net/
- De facto standard for implementing software
- Great design principles for refactors and green field
- Pros: Free; Up-to-date; Roadmap
- Cons: State has to exist somewhere; lightly addressed
- Quip: If apps only had 12 factors...

RELEASE IT!

- Developer centric cases and examples for releasing
- First edition out of print; second edition in December
- Technical af
- Pros: Looks at the SDLC holistically
- Cons: Not readily available yet
- Quip: Interesting that Release It! Isn't released yet.

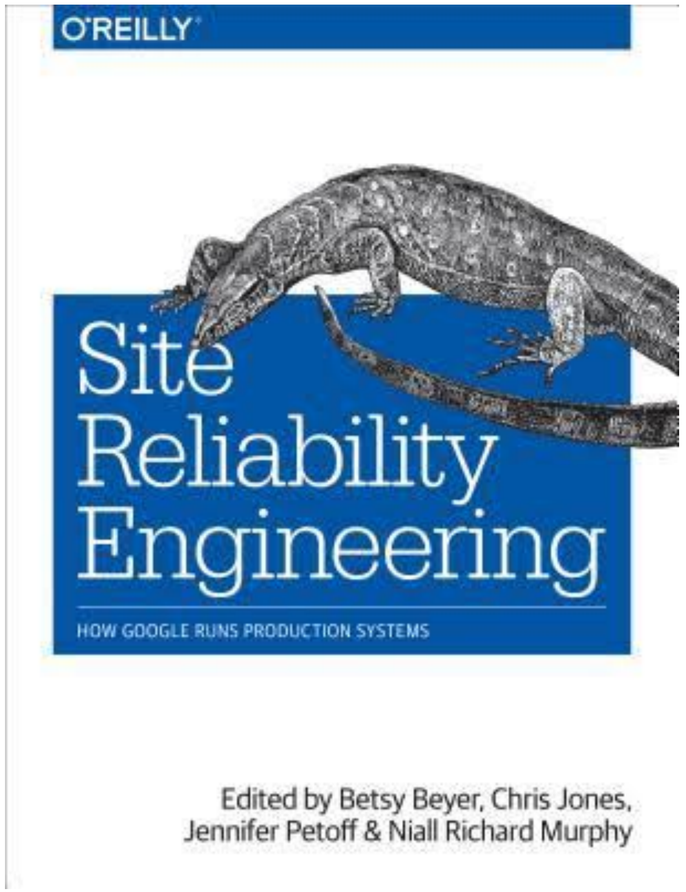
CONTINUOUS DELIVERY

- Provides focus for deploying software faster
- Emphasizes automation (you must automate first)
- When John Willis says "shift left" he's talking about executing earlier in pipelines described in this book
- Pros: Clear, real-world
- Cons: Sometimes redundant, slightly dated
- Quip: CD for your CTO to improve ROI and EBITDA.



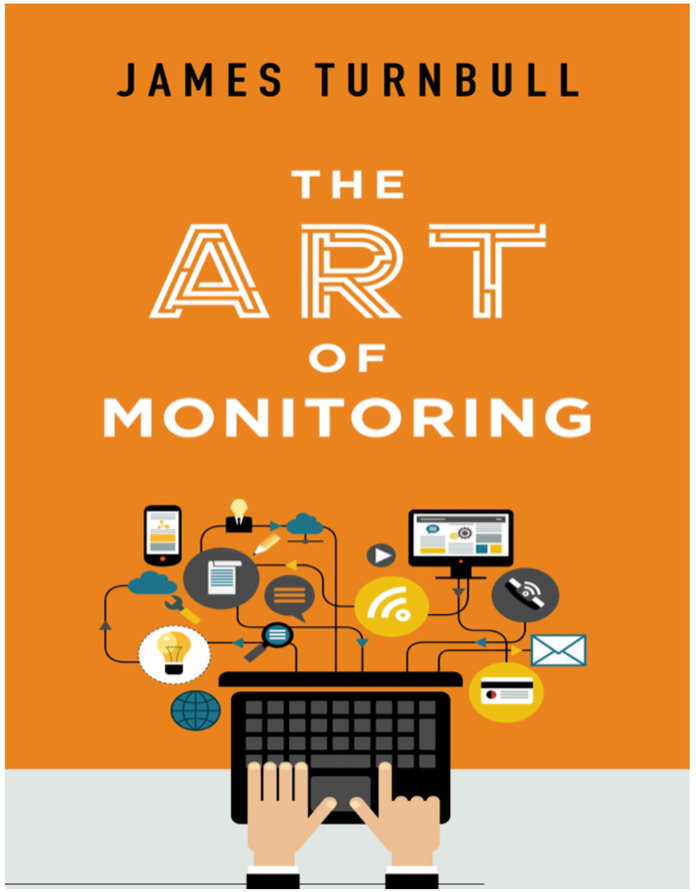
SITE RELIABILITY ENGINEERING

- A collection of essays from Google SREs about how things are done at Google
- A fantastic reference for various functions like on-call, onboarding, delivery, etc.
- Pros: Free; great examples of how to do things
- Cons: You are not Google; embrace with caution
- Quip: Google SRE is proof setting a pile of money on fire is a viable solution to engineering problems.



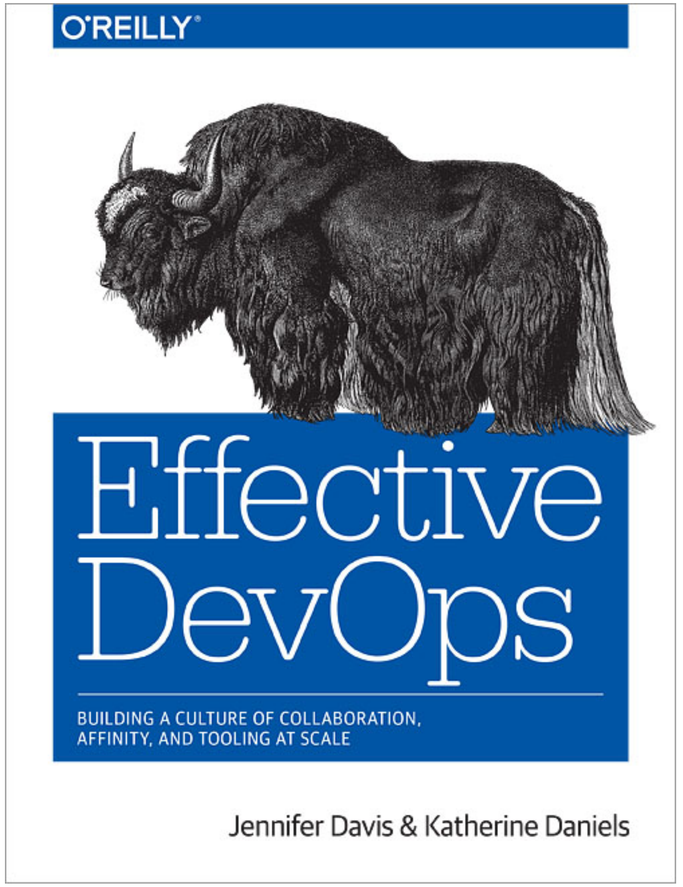
THE ART OF MONITORING

- Opinionated HOWTO implementation guide to monitoring at scale
- Incredibly thorough book
- Pros: Explicit; Detailed
- Cons: Opinionated; Long; Perhaps too specific
- Quip: If a book's art worthiness is measured by weight then we have a winner (767 pages).



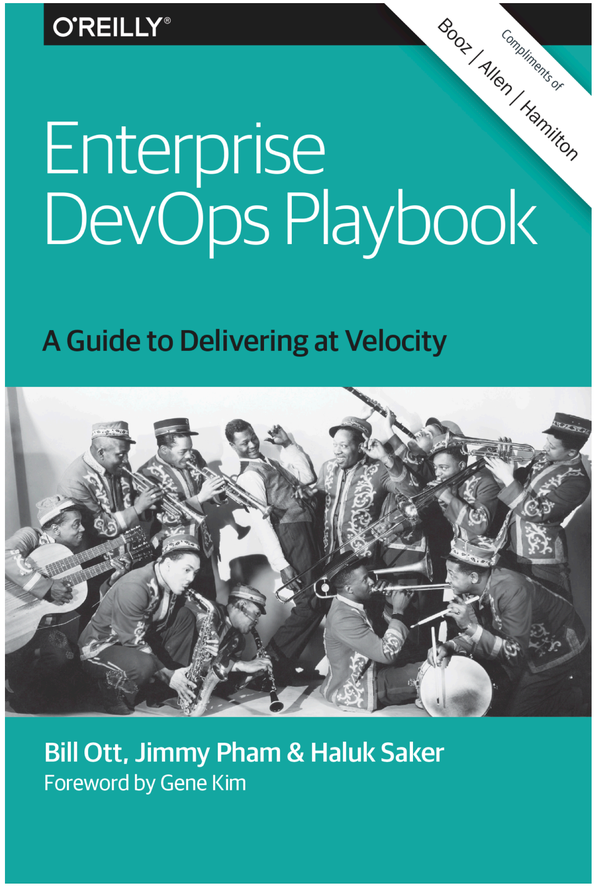
EFFECTIVE DEVOPS

- Culture centric focus on DevOps
- Discusses collaboration, hiring, team building, etc.
- Great for leaders and managers
- Touches on a wide variety of important topics
- Pros: Culture is hard; this helps
- Cons: Etsy probably isn't a great example anymore
- Quip: Effectiveness is a good thing!



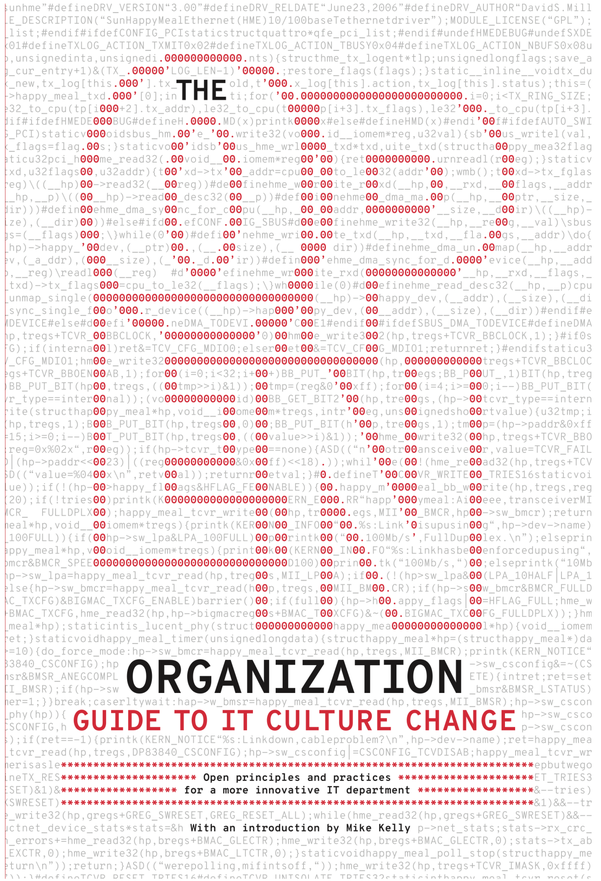
ENTERPRISE DEVOPS PLAYBOOK

- Roadmap for building a successful DevOps org
- Addresses hiring, culture, and learning
- Pros: Suggests tuning in your current organization
- Cons: Missing some pieces to the puzzle
- Quip: Yes! Enterprise and DevOps can work together... Somehow.



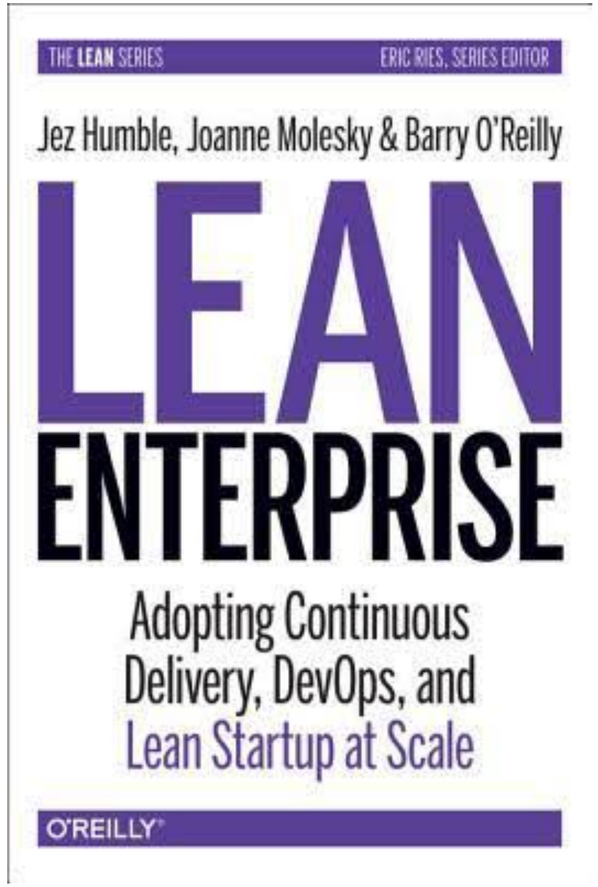
THE OPEN ORGANIZATION GUIDE TO IT CULTURE CHANGE

- Community-produced companion to Jim Whitehurst's 2015 book, The Open Organization
- Like SRE book, a collection of essays
- Focus on principles and practices of culture
- Pros: Easy to read; diverse authors; inspirational
- Cons: Not all pieces apply to everyone
- Quip: "Being positive is sometimes difficult to do." I wrote that? Wow. I had no idea.



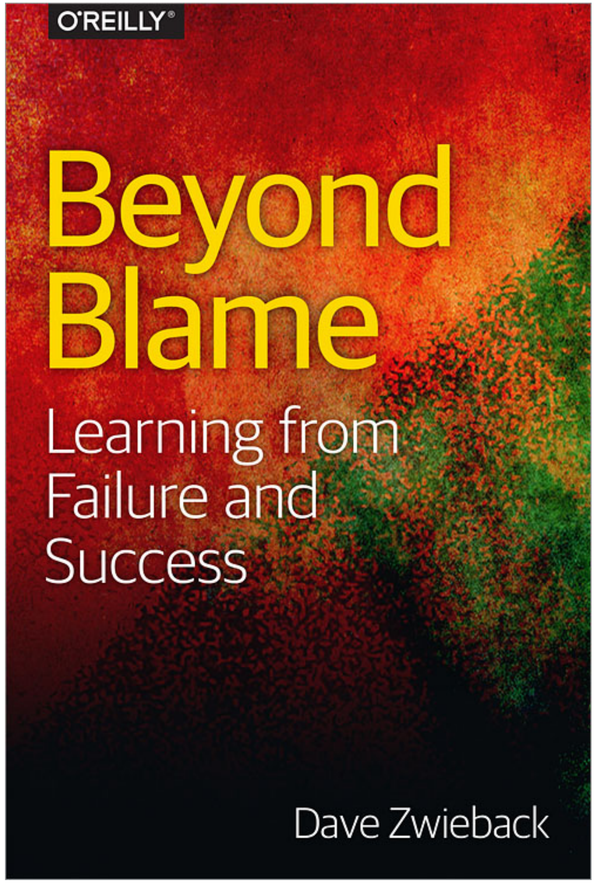
LEAN ENTERPRISE

- Big picture, business minded change agent
- All phase guide to planning, organizing, implementation, and measurement
- Great for leaders and managers
- Pros: Mindset changing readiness guide
- Cons: None given the scope
- Quip: This is not a weight loss book... Or is it?



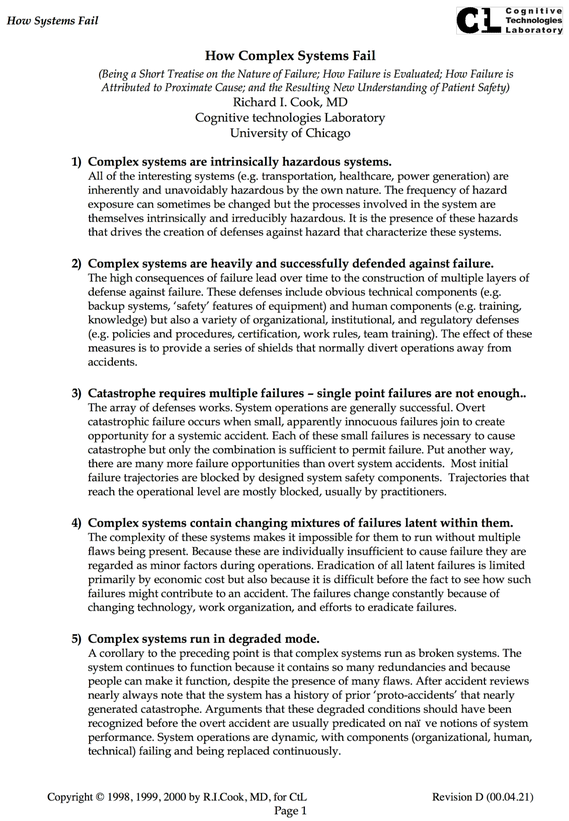
BEYOND BLAME

- Failure happens; Beyond Blame is a HOWTO in making postmortems blameless
- Great for individual contributors, leaders, managers
- Pros: Guides you towards blamelessness
- Cons: Emotions are hard, this isn't a psychiatrist
- Quip: I blame this book for your blame problems.



HOW COMPLEX SYSTEMS FAIL

- "Post-accident attribution accident to a ‘root cause’ is fundamentally wrong"
- Re-thinking failure in our systems makes them more robust
- Pros: Makes case that RCA isn't a solid process
- Cons: None given the scope
- Quip: You're human so you're the problem.



How Systems Fail

CtL Cognitive Technologies Laboratory

How Complex Systems Fail

(Being a Short Treatise on the Nature of Failure; How Failure is Evaluated; How Failure is Attributed to Proximate Cause; and the Resulting New Understanding of Patient Safety)

Richard I. Cook, MD
Cognitive technologies Laboratory
University of Chicago

1) **Complex systems are intrinsically hazardous systems.**
All of the interesting systems (e.g. transportation, healthcare, power generation) are inherently and unavoidably hazardous by the own nature. The frequency of hazard exposure can sometimes be changed but the processes involved in the system are themselves intrinsically and irreducibly hazardous. It is the presence of these hazards that drives the creation of defenses against hazard that characterize these systems.

2) **Complex systems are heavily and successfully defended against failure.**
The high consequences of failure lead over time to the construction of multiple layers of defense against failure. These defenses include obvious technical components (e.g. backup systems, ‘safety’ features of equipment) and human components (e.g. training, knowledge) but also a variety of organizational, institutional, and regulatory defenses (e.g. policies and procedures, certification, work rules, team training). The effect of these measures is to provide a series of shields that normally divert operations away from accidents.

3) **Catastrophe requires multiple failures – single point failures are not enough..**
The array of defenses works. System operations are generally successful. Overt catastrophic failure occurs when small, apparently innocuous failures join to create opportunity for a systemic accident. Each of these small failures is necessary to cause catastrophe but only the combination is sufficient to permit failure. Put another way, there are many more failure opportunities than overt system accidents. Most initial failure trajectories are blocked by designed system safety components. Trajectories that reach the operational level are mostly blocked, usually by practitioners.

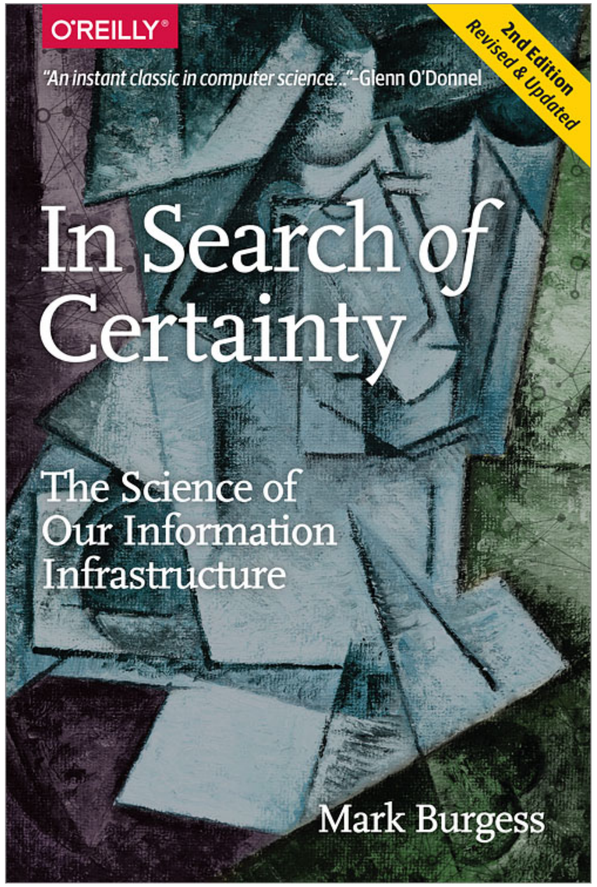
4) **Complex systems contain changing mixtures of failures latent within them.**
The complexity of these systems makes it impossible for them to run without multiple flaws being present. Because these are individually insufficient to cause failure they are regarded as minor factors during operations. Eradication of all latent failures is limited primarily by economic cost but also because it is difficult before the fact to see how such failures might contribute to an accident. The failures change constantly because of changing technology, work organization, and efforts to eradicate failures.

5) **Complex systems run in degraded mode.**
A corollary to the preceding point is that complex systems run as broken systems. The system continues to function because it contains so many redundancies and because people can make it function, despite the presence of many flaws. After accident reviews nearly always note that the system has a history of prior ‘proto-accidents’ that nearly generated catastrophe. Arguments that these degraded conditions should have been recognized before the overt accident are usually predicated on naïve notions of system performance. System operations are dynamic, with components (organizational, human, technical) failing and being replaced continuously.

Revision D (00.04.21)

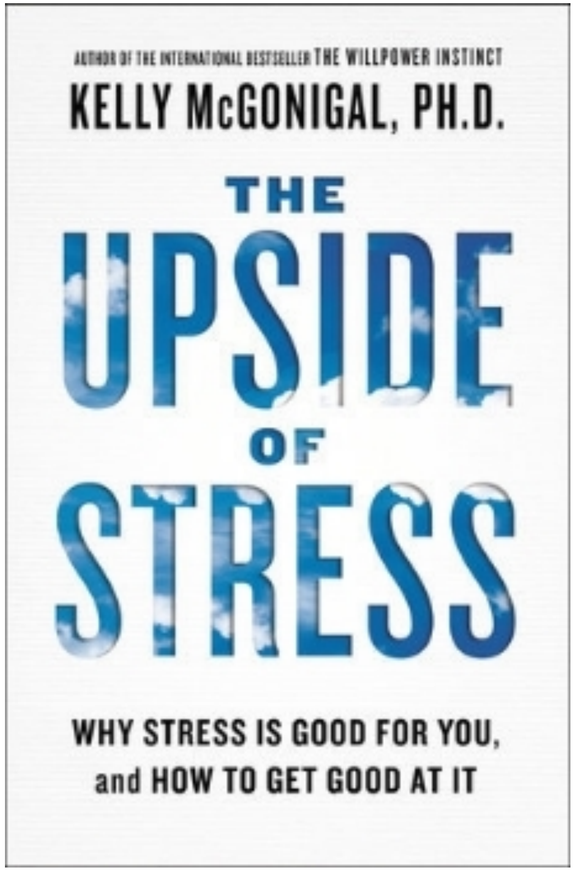
IN SEARCH OF CERTAINTY

- Foundation shaking look at future
- Great for individual contributors, leaders, managers
- Pros: Helps manage a world we don't know
- Cons: Slightly terrifying
- Quip: Death, taxes, and PagerDuty are the only certainties in life.



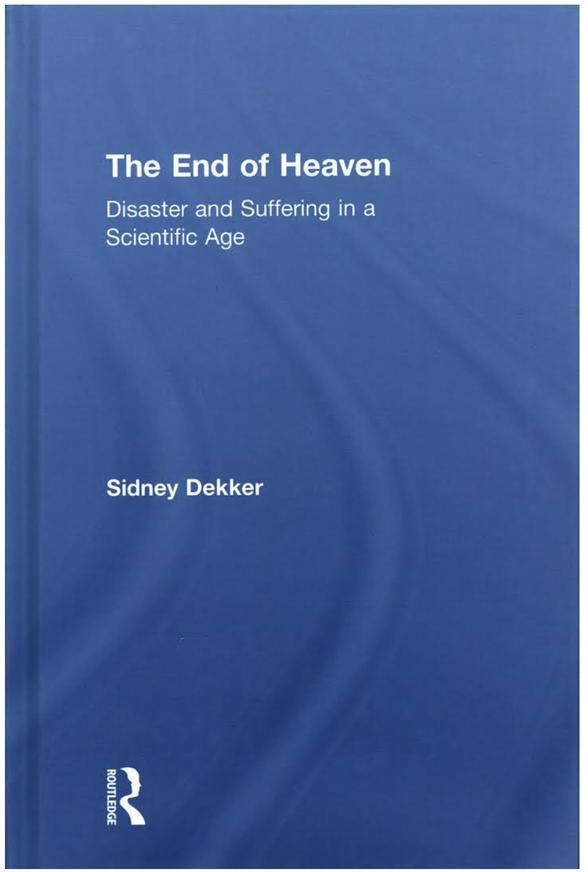
THE UPSIDE OF STRESS: WHY STRESS IS GOOD FOR YOU, AND HOW TO GET GOOD AT IT

- Stress isn't all bad if we learn how to manage it
- Stress can actually make us happier
- Pros: Teaches life improving skills
- Cons: None given the scope
- TED Talk: How to make stress your friend
- Quip: If stress is good for me I'm going to live forever.



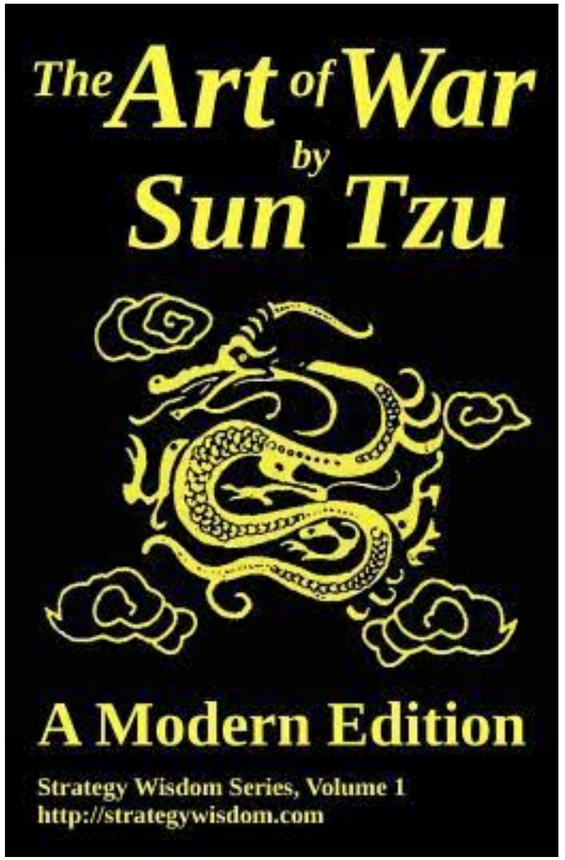
THE END OF HEAVEN: DISASTER AND SUFFERING IN A SCIENTIFIC AGE

- Slightly controversial take on disaster in modern times
- Recommended to me by John Willis
- Pros: Makes you rethink your feelings
- Cons: Slightly controversial



THE ART OF WAR

- In DevOps you SHOULD NOT have adversaries
- I am willing to bet that anyone worth their salt has read this though
- Tactics from this work should be used sensibly
- "Know thy enemy"
- Pros: Well known work studied in business, military
- Cons: Not an easy read; multiple differing translations



YOU ARE EITHER BUILDING A LEARNING ORGANIZATION OR YOU WILL BE LOSING TO SOMEONE WHO IS.

Andrew Clay Shafer