



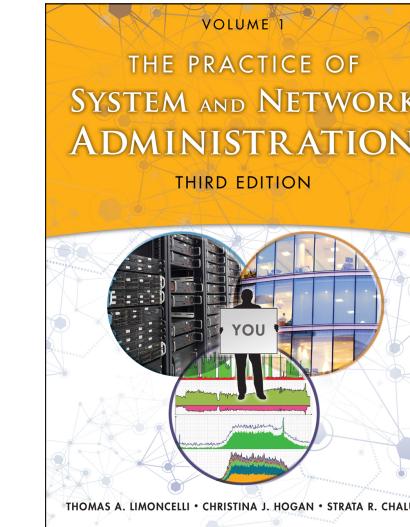
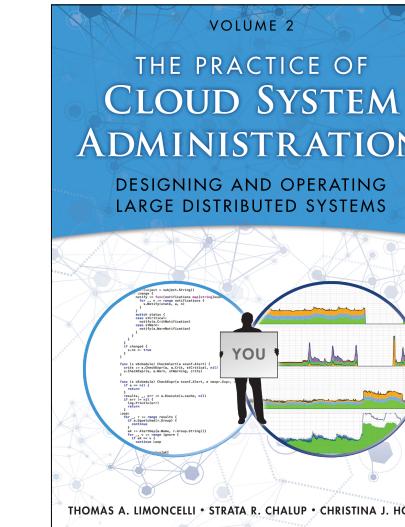
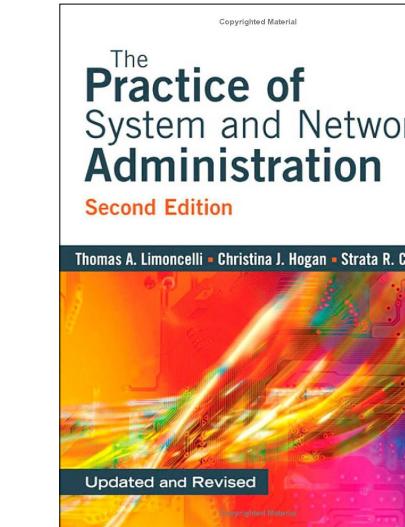
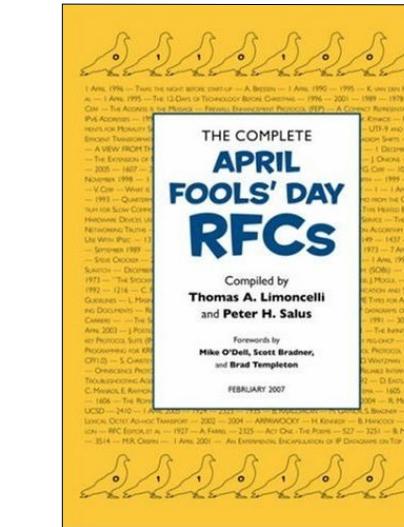
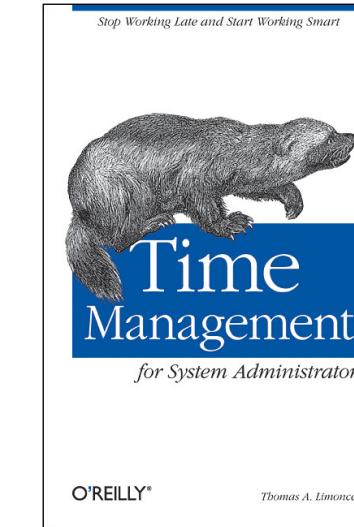
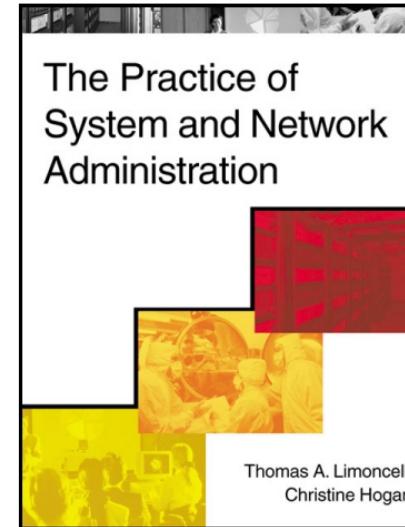
Operations Reform: Tom Sawyer-ing your way to better operations

Thomas A. Limoncelli
SRE Manager
@YesThatTom



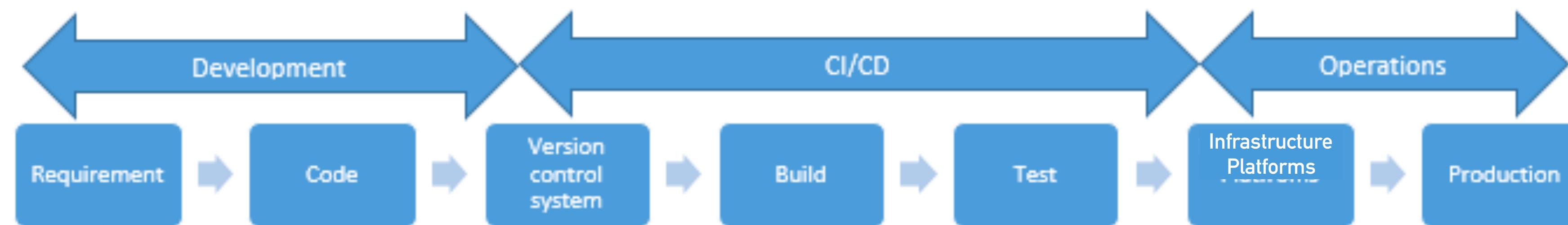
Who is Tom Limoncelli?

- SRE Manager at Stack Overflow, Inc (NYC)
- Sysadmin since 1988 (Google, AT&T/Bell Labs, Mentor)
- Blog: EverythingSysadmin.com Twitter: @YesThatTom
- 



Most Talks

This talk



Three Stories

Story 1:

The Big Initiative

What I learned

- Operations teams can say “no” to “Top-Down” edicts.
 - They’re under-funded. Don’t have the resources.
 - They’re overloaded. Don’t have the time.
 - There’s too much history. Complexity is killing them.
- Therefore... I always look for alternatives to Top-Down edicts.



The screenshot shows a web browser window with the following details:

- Address Bar:** https://itrevolution.com/book/expanding-pockets-greatness/
- Page Title:** Expanding Pockets of Greatness
- Header:** Includes links for COLLABORATORS, ABOUT, and CONTACT, along with a search bar.
- Navigation Bar:** Includes links for BOOKS, FORUM PAPERS, RESEARCH, EVENTS, SPEAKING, and BLOG.
- Content Area:**
 - Section Header:** EXPANDING POCKETS OF GREATNESS
 - Text:** Spreading DevOps Horizontally in Your Organization
 - Section Header:** FREE DOWNLOAD
 - Text:** How do you build momentum and go from a few islands of DevOps goodness to a tipping point where the entire organization embraces common DevOps methods? This paper provides guidance to leaders (change agents) who see DevOps activity in different teams, groups, or departments in their organization and are looking to build community and promote growth. This paper will help you connect the pockets of DevOps goodness that may already exist within your organization and harness them to drive change.
- Call-to-Action:** DOWNLOAD THIS FREE RESOURCE AND START TRANSFORMING TODAY! (with a blue button labeled DOWNLOAD)

Story 2:

Google's

Operational Hygiene

Assessment

Service A	12/2014	1/2015	2/2015	3/2015	4/2015
Regular Response	1	2	2	2	3
Emergency Response	3	3	4	4	4
Monitoring and Metrics	1	1	2	2	2
Capacity Planning	1	1	1	1	2
Life-Cycle Management	2	2	2	3	2
New Service Introduction and Removal	1	2	2	3	3
Service Deployment and Decommissioning	2	2	3	3	4
Resource Efficiency	1	1	1	2	2

Team 1	2014Q1	2015Q1	2015Q2	2015Q3	2015Q4
Service A	1	2	2	2	1
Service B	3	4	4	4	5
Service C	1	2	2	3	3

	2014Q1	2015Q1	2015Q2	2015Q3	2015Q4
Team 1	1	2	2	3	3
Team 2	3	4	3	4	5
Team 3	1	1	1	1	2
Team 4	1	1	2	2	2
Team 5	2	3	3	3	4

Why this worked?

- **Psychologically:**
 - Simplicity: A spreadsheet per team, linked like crazy.
 - Low barrier to entry: Easy to get started.
 - Leverages pride and ego as motivators.
- **Creates good culture:**
 - Blameless: Assess the service, not the people.
 - Transparency and responsibility: A culture of fixing things, not hiding things.
- **Non-monetary recognition of good work:**
 - Encourages copying greatness: Unsure how to improve? Talk to (and copy) other people's success.
 - Outstanding work recognized by peers.
- **Helps direct cross-team impact.**
 - Easy to identify larger needs.

Why this worked?

- **Psychologically:**
 - **Simplicity:** A spreadsheet per team, linked like crazy.
 - **Low barrier to entry:** Easy to get started.
 - **Leverages pride and ego as motivators.**
- **Creates good culture:**
 - Blameless: Assess the service, not the people.
 - Transparency and responsibility: A culture of fixing things, not hiding things.
- **Non-monetary recognition of good work:**
 - Encourages copying greatness: Unsure how to improve? Talk to (and copy) other people's success.
 - Outstanding work recognized by peers.
- **Helps direct cross-team impact.**
 - Easy to identify larger needs.

Why this worked?

- **Psychologically:**
 - Simplicity: A spreadsheet per team, linked like crazy.
 - Low barrier to entry: Easy to get started.
 - Leverages pride and ego as motivators.
- **Creates good culture:**
 - **Blameless: Assess the service, not the people.**
 - **Transparency and responsibility: A culture of fixing things, not hiding things.**
- **Non-monetary recognition of good work:**
 - Encourages copying greatness: Unsure how to improve? Talk to (and copy) other people's success.
 - Outstanding work recognized by peers.
- **Helps direct cross-team impact.**
 - Easy to identify larger needs.

Why this worked?

- **Psychologically:**
 - Simplicity: A spreadsheet per team, linked like crazy.
 - Low barrier to entry: Easy to get started.
 - Leverages pride and ego as motivators.
- **Creates good culture:**
 - Blameless: Assess the service, not the people.
 - Transparency and responsibility: A culture of fixing things, not hiding things.
- **Non-monetary recognition of good work:**
 - **Encourages copying greatness: Unsure how to improve? Talk to (and copy) other people's success.**
 - **Outstanding work recognized by peers.**
 - **Helps direct cross-team impact.**
 - Easy to identify larger needs.

Why this worked?

- **Psychologically:**
 - Simplicity: A spreadsheet per team, linked like crazy.
 - Low barrier to entry: Easy to get started.
 - Leverages pride and ego as motivators.
- **Creates good culture:**
 - Blameless: Assess the service, not the people.
 - Transparency and responsibility: A culture of fixing things, not hiding things.
- **Non-monetary recognition of good work:**
 - Encourages copying greatness: Unsure how to improve? Talk to (and copy) other people's success.
 - Outstanding work recognized by peers.
- **Helps direct cross-team impact.**
- **Easy to identify larger needs.**

Non-monetary recognition of good work:

If low scores resulted in low bonuses, the best people wouldn't swarm in to fix the failing services.

- Create a virtuous cycle: Reward improvement, not state
- You want to encourage your best people to leave high-assessed teams and join low-assessed teams.
 - They have the skills and (more importantly) experience
 - They won't move if it means a pay cut

Seek perfection. Don't require it.

Demanding perfection creates an incentive to lie.

- Perfection means you've wasted money.
 - The last 10% costs more than the other 90%
 - Focus your time on the important 90%

Story 3:

Stack Overflow's Operational Hygiene Assessment

SO is a lot smaller

- One SRE team, many responsibilities.
- More granular definition of “service”
- Scaled the process down:
 - One spreadsheet
 - SREs made their own rubric
 - Scores were pass/fail

Scoring

A	Pass
B	Fail

Scoring

A	Pass
B	Fail
F*	Fail (OMG!)

Scoring

A*	Pass (Cutting Edge! Exemplar!)
A	Pass
B	Fail
F*	Fail (OMG!)

A	B	C	◀ ▶	E	F	G	H	I	J	K	
S3 Assessment	vTeam	Service		Backups	Upgradability	Failover/Resilien	Monitoring	Build Automation	Dev/Test areas	Documentation	Se
Traffic											
Fastly		A	A	A	A	A	A	A	A	A	A
haproxy (lb)		A	B	A	A	A	A	A	B	A	A
IIS servers		B	B	A	A	B	B	A	B	B	B
fastly-ratelimitor		A	A	F	A	F	A	B	B	A	A
TPS		A	A	A	A	B	A	B	B	A	A
dnscontrol		A	A	A	A	B	A	A	A	A	A
Logstash/Kibana fastly-*, trafficlogs-*, arn/a		n/a	B	F	A	B	A	B	B	n/a	F
GoogleDNS/Route53		A	A	A	B	A	A	A	n/a	A	A
TLS certs (digicert)		A	F	n/a	A	A	B	A	A	A	A
AD DNS		A	B	A	B	F	B	B	B	B	B
Cloudflare		A	A	A	B	A	A	A	A	A	A
registrars		n/a	n/a	n/a	n/a	n/a	n/a	n/a	A	A	A
Pingdom		B	B	B	B	F	B	B	B	A	A
Pagerduty		n/a	n/a	B	B	F	B	B	B	A	A
puppet-based bans		A	A	n/a	B	A	B	A	A	A	A
Database	Database Instance Groups										
	Core/SO	A	B	A	A	B	A	B	B	B	B
	SE Network	A	B	A	A	B	A	A	B	B	B
	ML SQL	A	A	A	A	B	n/a	B	B	B	B
	LogSQL database	n/a	A	A	A	B	A	B	B	B	B
	Channels	A	B	A	A	B	B	B	B	B	B

Assessment Categories

- **First Iteration:**
 - Backup & Restore
 - Upgradability
 - Failover/Resiliency
 - Monitoring
- **Added later:**
 - Build Automation
 - Separate Dev/Test areas
 - Documentation
 - Security
 - Capacity Planning

Why this worked?

- Simple. Simple Simple.
- Blameless. Assess the service, not the people.
- People motivated to expose their warts, and fix them.
- Invisible to management no longer!

“Don’t tell ME what’s wrong!
I know what’s wrong! Nobody listens!”

Created a new problem

- Identified dozens (hundreds) of new tech-debt projects.
- Temptation to ignore feature projects.

3 Solutions

1. Rate limit

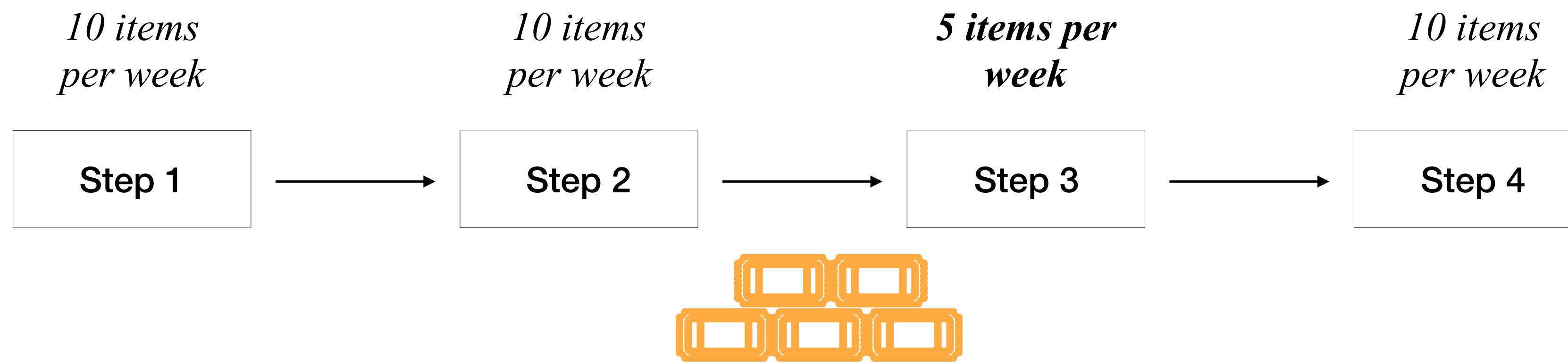
- Set goal of 20% project hours on tech debt

2. “Theme Month”

- “September is Fix Backups Month”

3. The Theory of Constraints

The Theory of Constraints



Three Stories

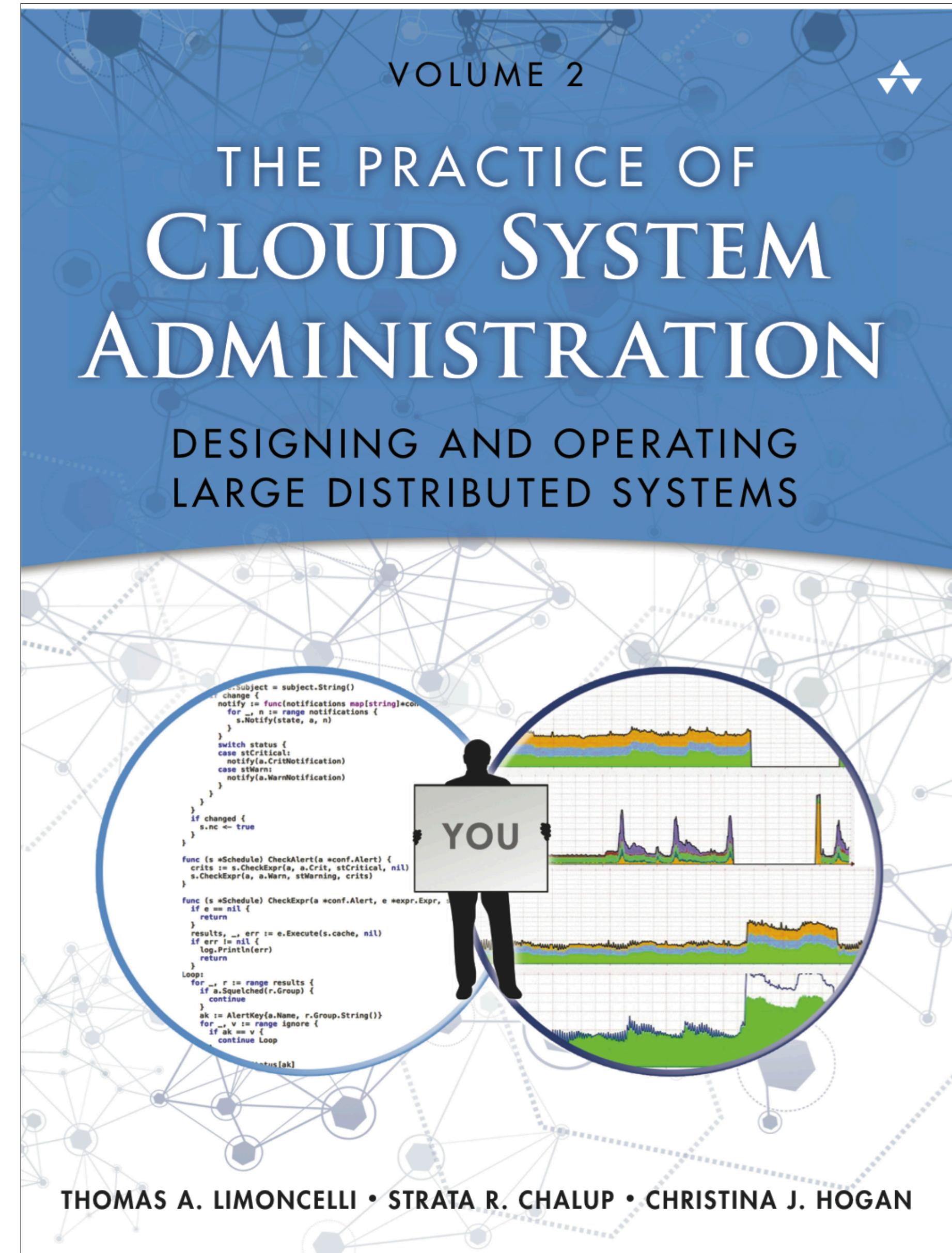
The End

Story 4:

DIY

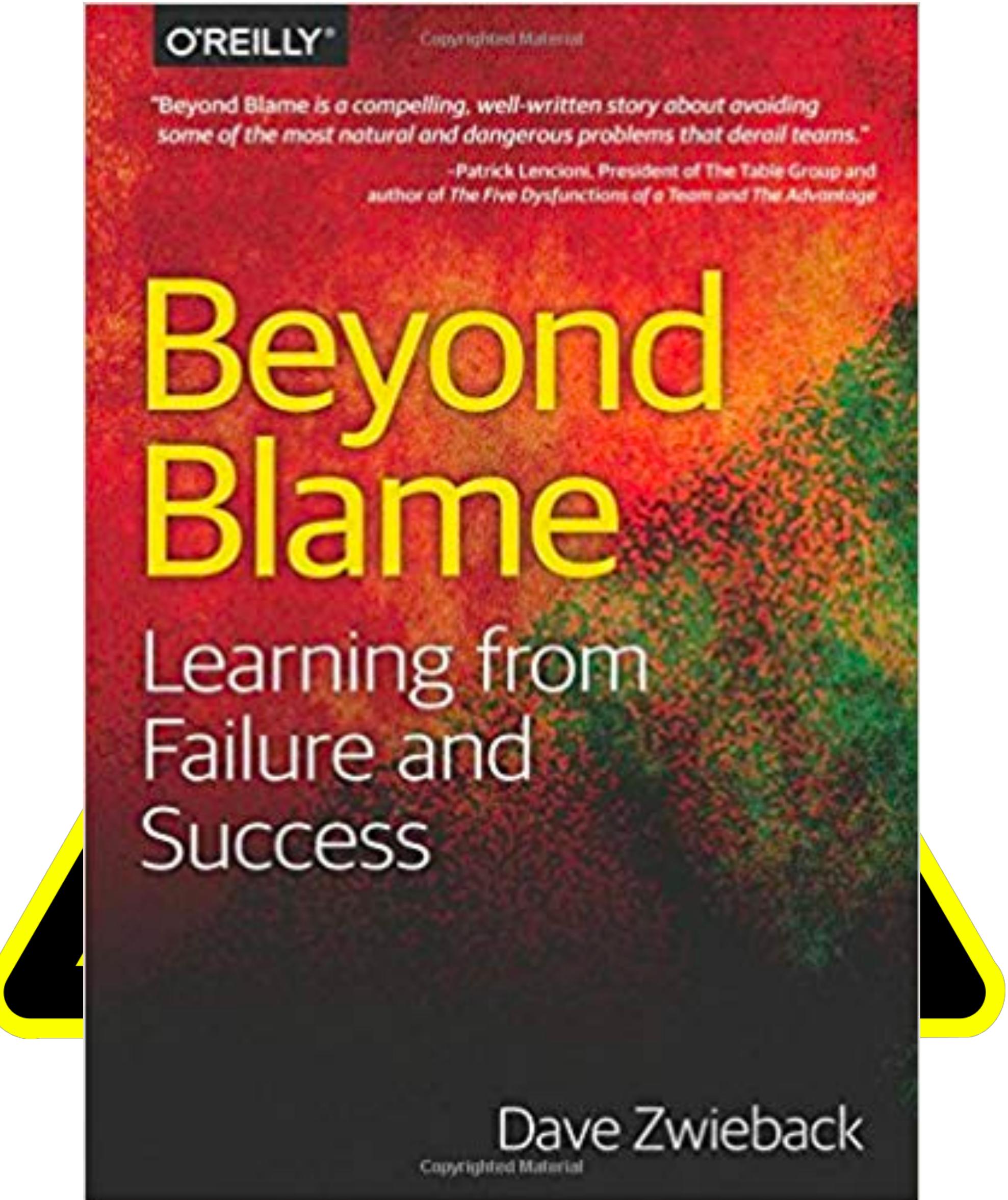
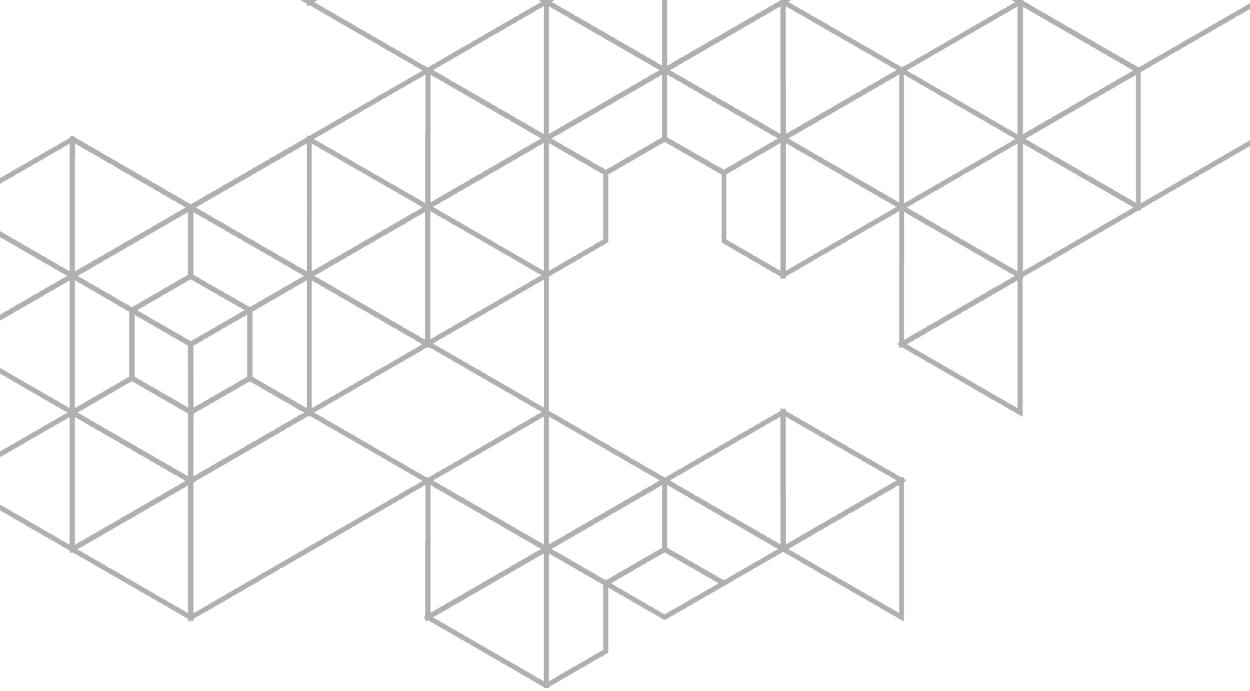
Chapter 20: The instruction manual

Appendix A: Sample assessment questions and “look for’s”



Do this in your enterprise

- Grass roots:
 - Management sets high standards, engineers figure out how to get there.
 - Let teams create their own rubrics, management silently keeps everyone to high standards.
- Start with 1 team, no “global initiative”
 - Grow to more teams over time.
- Rubric evolves over time
 - Let teams create/evolve rubric (job of senior members?)
- Provide resources; don’t just expect this to happen for free!
- Use a spreadsheet. Software development not needed.
- Create a blameless, transparent, assessment.





Thanks!

Any questions?

Chapter 20: The instruction manual

Appendix A: Sample assessment questions and “look for’s”

