

Transactional System Administration is Killing Us and Must be Stopped

Thomas A. Limoncelli, SRE

Stack Overflow

@YesThatTom EverythingSysadmin.com

usenix

LISA15

November 8–13, 2015 | Washington, D.C.

www.usenix.org/lisa15

#lisa15

Talk Outline

1. What is Transactional System Administration?
2. Why is it bad?
3. What should we do instead?

Who is Tom Limoncelli?

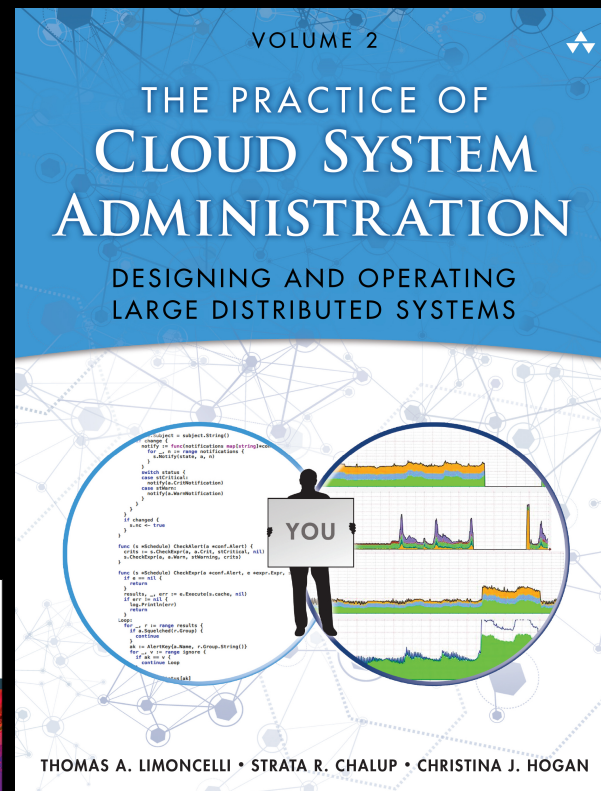
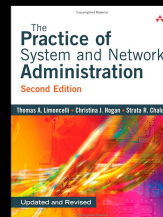
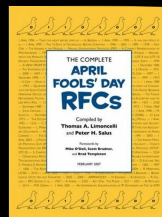
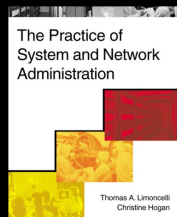
Sysadmin since 1988

Worked at Google, Bell Labs & many smaller companies.

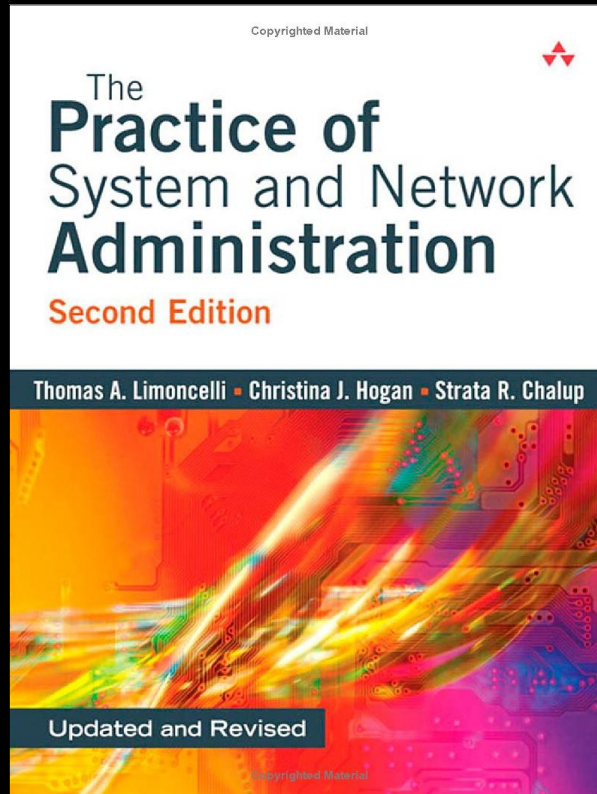
SRE at Stack Exchange, Inc
serverfault.com / stackoverflow.com

Blog: EverythingSysadmin.com

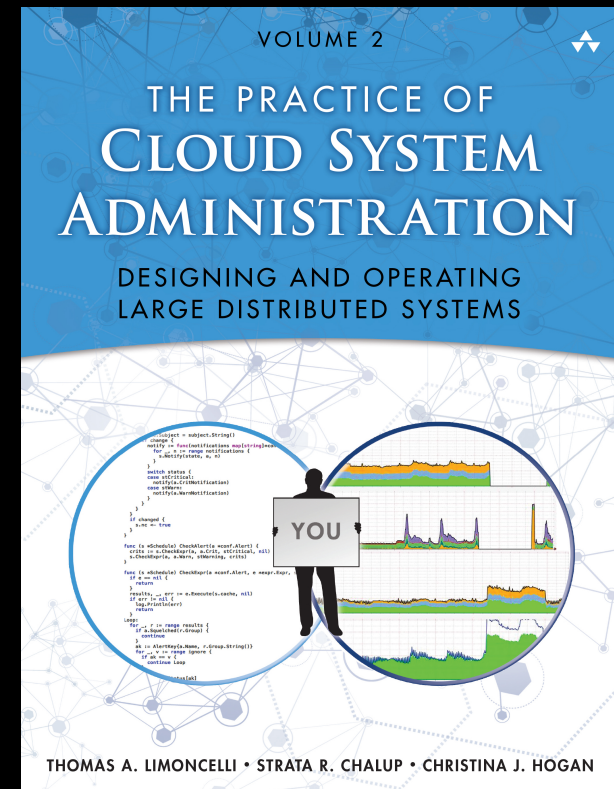
Twitter: [@YesThatTom](https://twitter.com/YesThatTom)



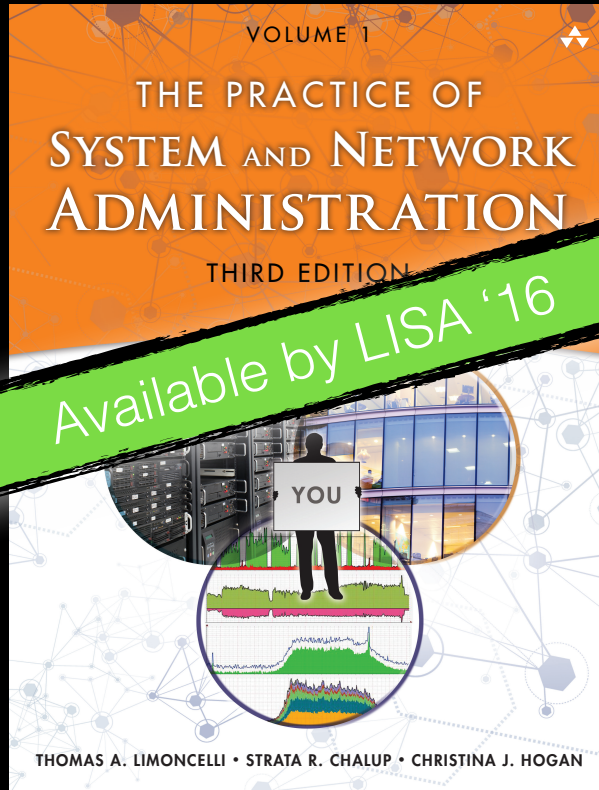
Volume 1



Volume 2

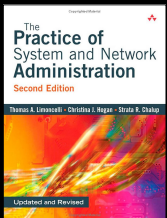


Volume 1

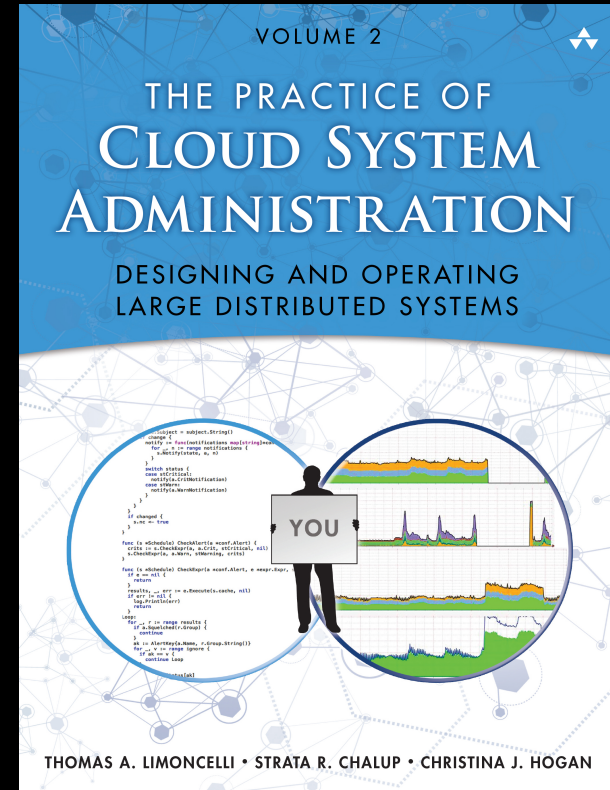


THIRD EDITION

Available by LISA '16



Volume 2



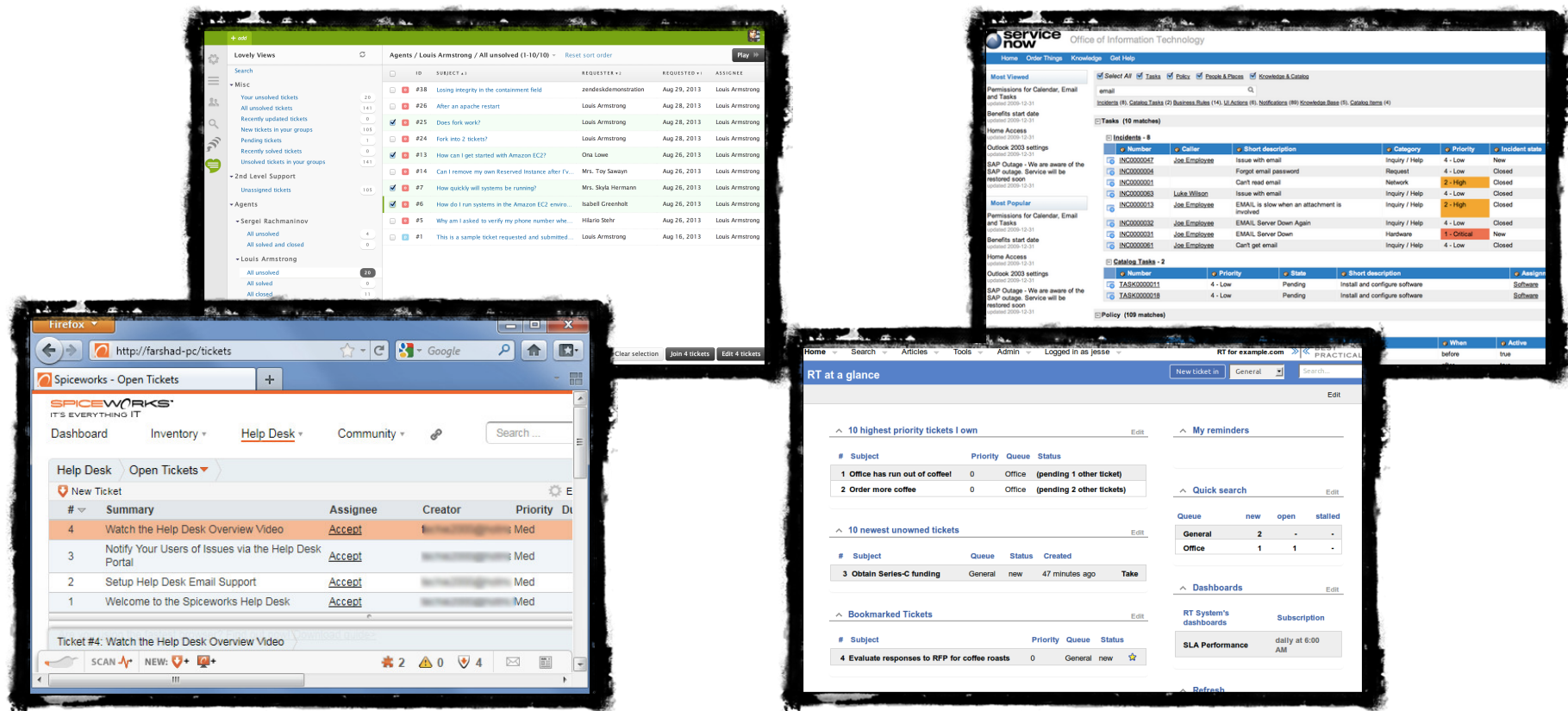
300+ pages of new material.
Updated and reorganized!

What is Transactional System Administration?

Request / response

- Customer: Would you please do x-y-z?
- System Administrator: Yes. Done.
- Customer: Verified.

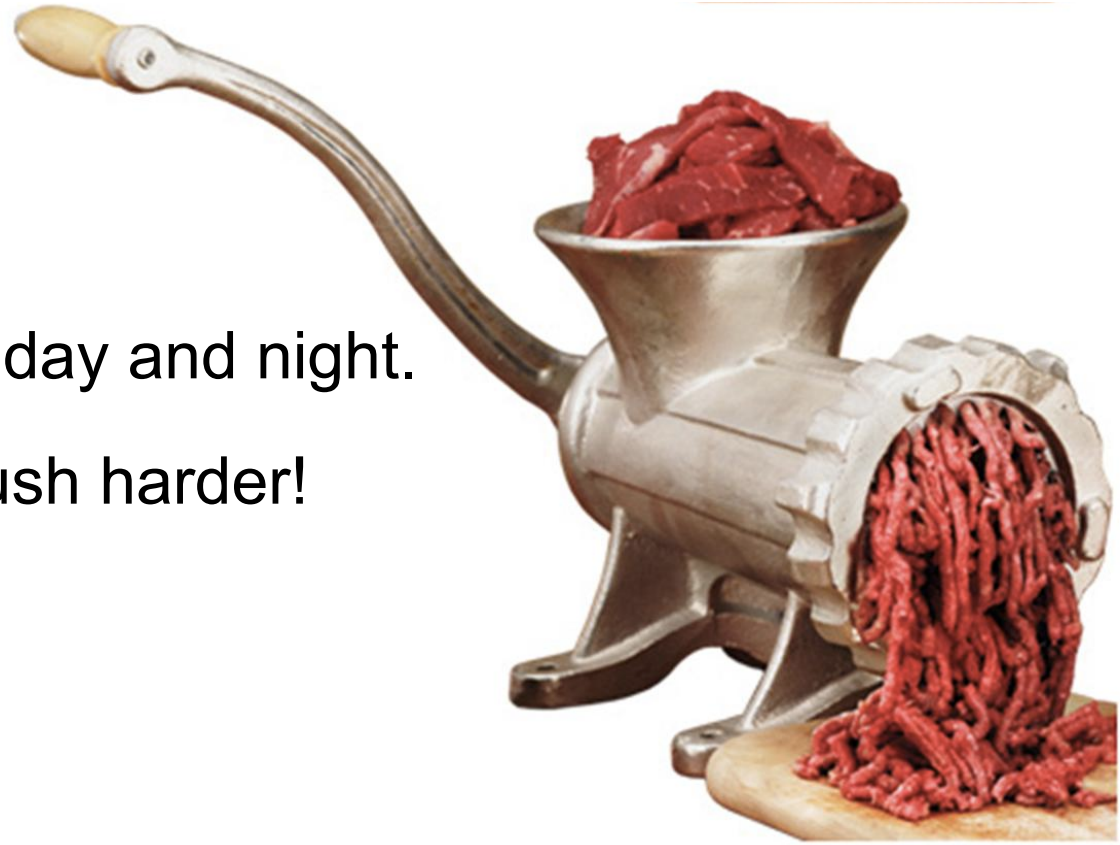
What does it look like?



Why is Trans SA bad?

Meat Grinder System Administration

- We grind through requests day and night.
- Need to process more? Push harder!



Need to process more requests?

- The problem with “Push harder!” is that the pressure is all on the system administrators.
- We have a finite amount of time.



Interrupt Driven

- It's reactive, not pro-active:
- Attention goes to the loudest customer
- Fire fighting ignores your good customers, teaches them to be the bad customer that yells the loudest!
- Soon everything is an emergency.

Discourages Long-Term Planning

- No capacity planning:
 - Nobody ever files a ticket requesting that you plan ahead.
 - Outages and emergency purchases.
- Doesn't (automatically) include optimizations.
 - No “built in” force to encourage automating top requests
- New service? Operations is an after-thought.

Creates the wrong power relationship

- Customers view Sysadmin as “servants”
- Sysadmins view customers as “PITA demanding children”
- Encourages “us vs. them” thinking

None of this is good.

Scaling By Hiring More People Is Bad

- **Expensive:** The opposite of “economies of scale”
- **Impossible:** Skills shortage (careers.stackoverflow.com)
- **Difficult:** Managing $2n$ people is n^2 more difficult
- **Antiquated:** It is so 1990s!
 - See email thread "**How many sysadmins?**" **redux** on the `lisa-members` mailing list

Oh yeah, users hate creating tickets!

- Intimidating! (embarrassing to not know the right words)
- Painful! (users struggle to write, speaking is easier)
- Annoying! (has no “instant feedback”)
- Ineffective! (“like sending requests into a black hole”)

10 tips ways to make it easier
for users to open tickets.

4 ways to train users to
communicate via tickets

6 tips to get users to open tickets instead
of going directly to your desk

MAYBE TICKETS ARE THE PROBLEM

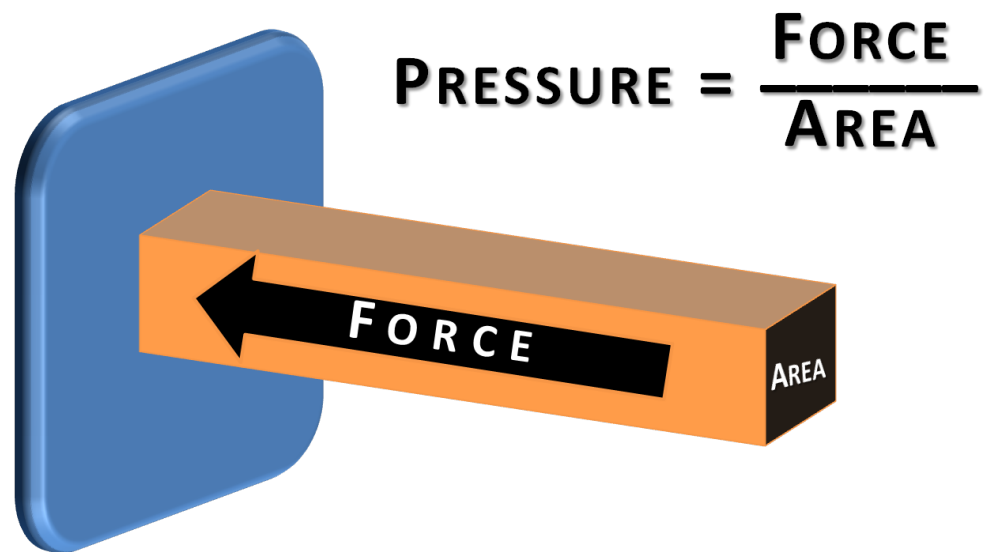
6 ways to organize
your tickets better

3 ways to make their tickets an
el
ored

Is Transactional System
Administration
really “killing us”?

Yes!

- High Pressure
- High Stress



Depression, Burn-Out, Mental Health

- Karōjisatsu
- By John Willis

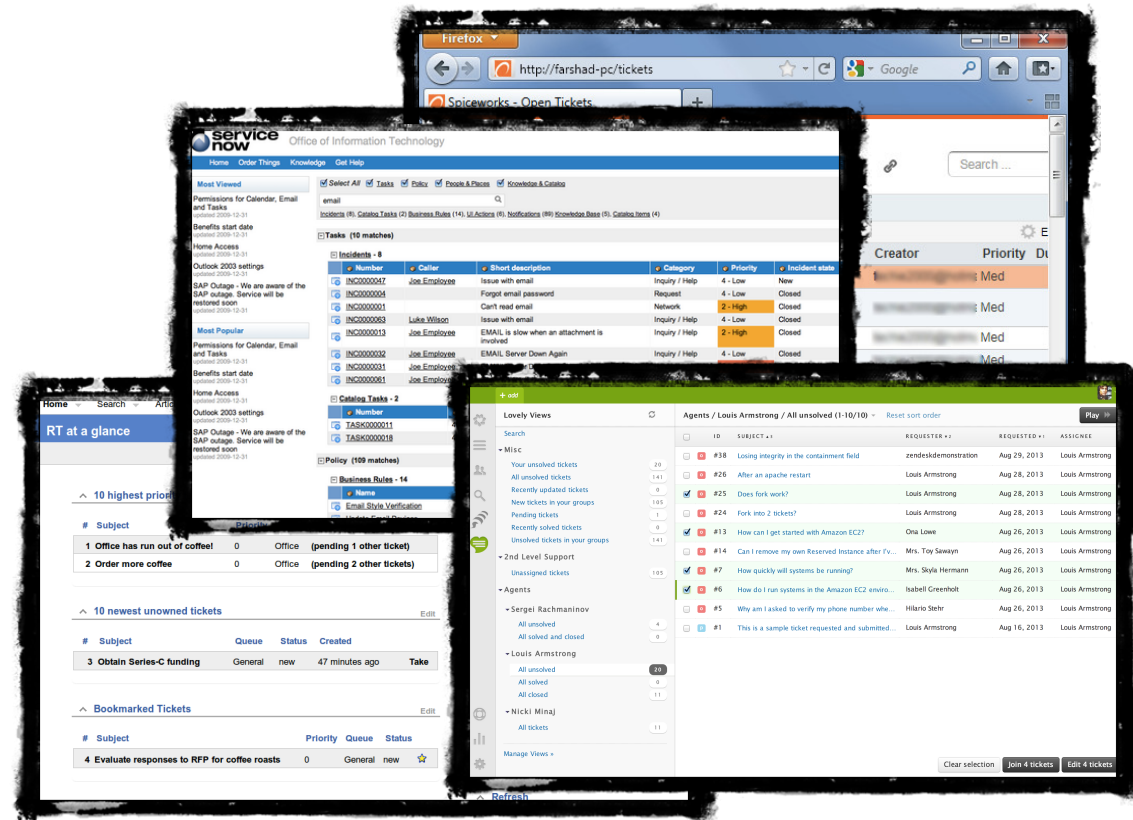


“An Honest Job Advert”

How did we get here?

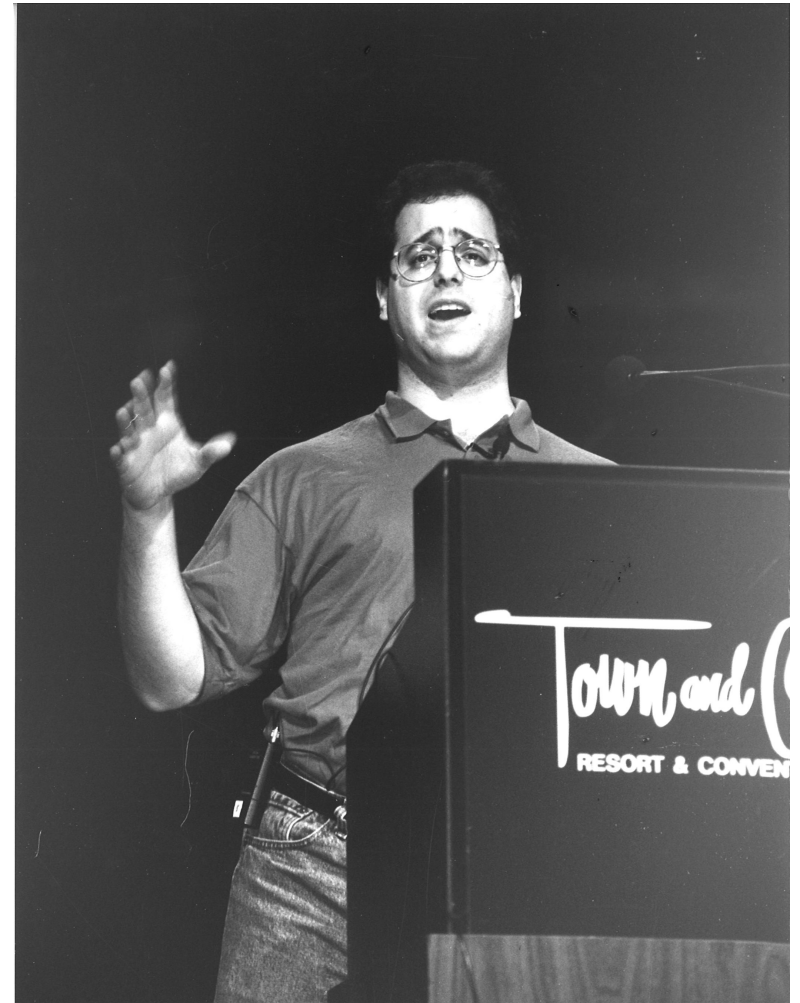
Too much emphasis on Ticket Systems

- Better than total chaos
- But a curse nonetheless

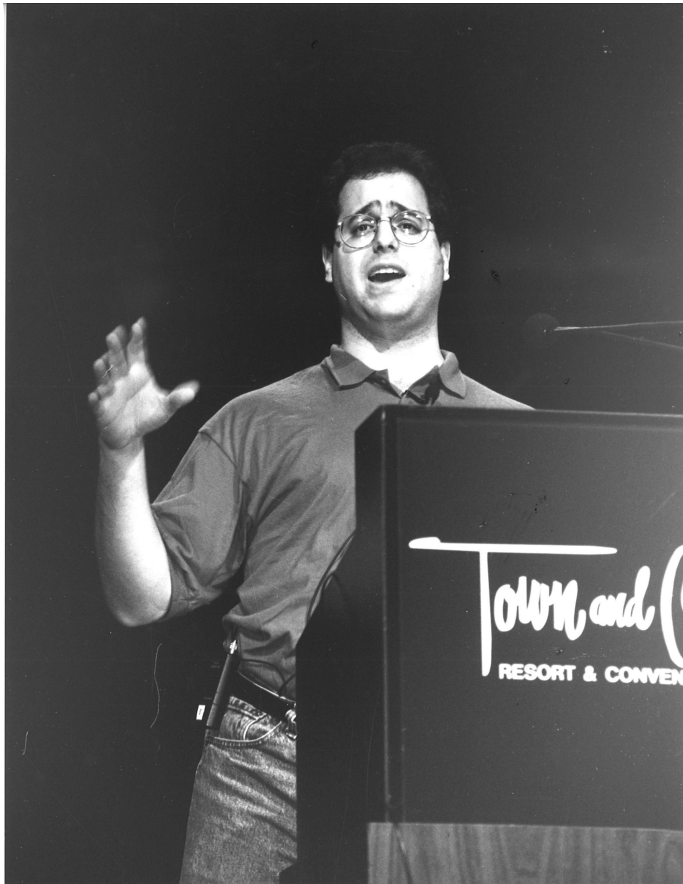


I was part of the problem

- Started advocating that people adopt “helpdesk automation” in 1991



Me presenting at LISA '97



1997



1999



“Helpdesk Automation”

- PUSH HARDER!



Maybe it's our fault?

(No, it isn't)



You need a way to track you work.
(but it shouldn't dictate your life)

What to do instead?

Three Examples

1. Stack Overflow
2. Google SRE
3. Kanban

Example 1: Stack Overflow

StackOverflow: Collaborative SRE

- Killed the ticket system
- Each Dev Team gets one SRE representative.
 - Rep attends the Dev Teams meetings, chat room, etc.
- Ops would assigned to SRE just like any other dev task:
 - "Add menu item for xyz"
 - "set up new replica"

SRE is a Dev-team member that does the ops-related tasks

Results so far?

- Push -> Collaboration
- Still new but working ok.
- Not all teams have a representative. Those teams get help on-demand by speaking up in the SRE chat room.
- Only works because we are small.

Example 2: Google SRE

Google SRE: Self-Service Tools

- SRE create “self-service” versions of the things Devs normally request sysadmins do:
 - Configure load balancer to point at service replicas
 - Compile software and build packages
 - Detect hard disk failures and replace

Monitoring

- SREs don't create monitoring rules for Devs.
- SREs created and run a system that lets Devs be self-sufficient for their monitoring needs
- Better because:
 - Devs have better knowledge of internals
 - “Workforce multiplier”

Ganeti

- ORIGINAL: Open Ticket. VM created manually.
- BETTER: Automated VM creation
- EVEN BETTER: Web-based portal.
- BEST: An API.

Handling failed disks

- Gmail has a lot of storage. Many disks on many machines.
- Disk failure likely every 100,000 hours.
- Detect Failure -> Drain -> Replace disk -> Reintegrate

Karl

eKarl

Fully automated

Laptop Distribution

- Sysadmins shouldn't install and deliver laptops to users.
- We should maintain the system that:
 1. Knows when a new laptop is needed:
 - Tracks HR database for new employees
 - Emails users when eligible for refresh
 - A portal for special ordering machines
 2. Automates OS installation for use by a “laptop delivery crew” in each building. (Non-technical clerks and IT coordinator)
 3. Does capacity planning, generates purchase orders, etc. for the purchasing/finance group.

This is better because...

- Scales better
 - As company grows; As # of locations grows
- Division of labor
 - Better use of your time
 - Uses less expensive labor as you scale up
- Less boring work for the sysadmins!



Don't build cars. Be the people that build the robots that build cars. You will never be obsolete.

Related example: Account Creation

- System monitors HR database for add/delete/changes.
- Creates/suspends/updates accounts.
- Sysadmins are out of the “account business”. They maintain the account creation system.

Service Administration

- You don't process transactions.
 - You run the system that processes transactions
- Better because:
 - The “pressure is on the service, not the people”
 - Need more volume? Scale the service.
 - Need faster service? Optimize the service
 - Better service for customers? Add/fix features
 - Less boring for the sysadmins

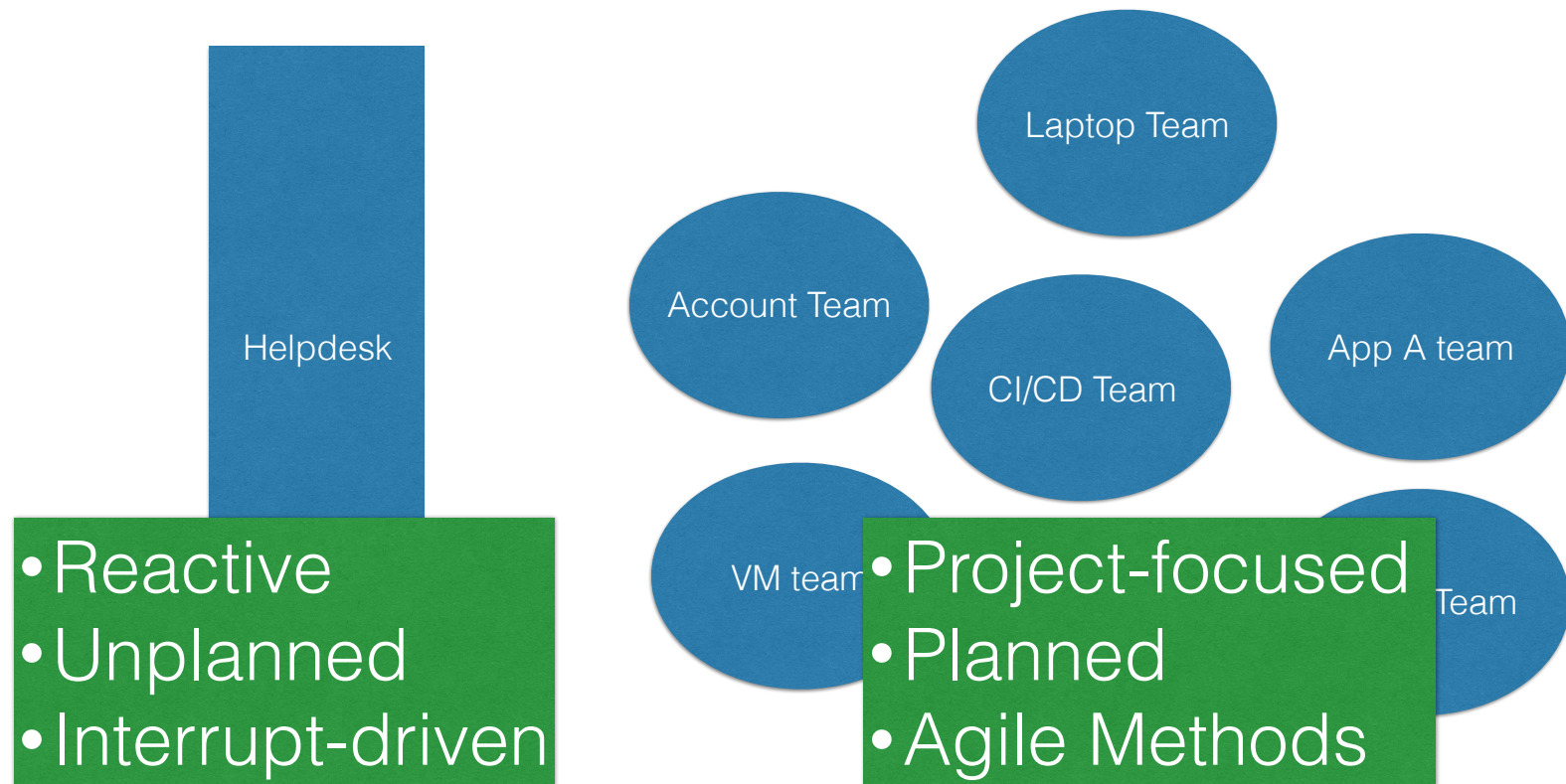
"If a human operator needs to touch
your system during normal
operations, you have a bug.
The definition of normal changes as
your systems grow."

Carla Geisser, SRE, Google

Summary:

- Push -> Software Development
 - People are now 1 degree away from the process.
 - The "push" is on the software, not the people

Helpdesk vs. Sysadmin

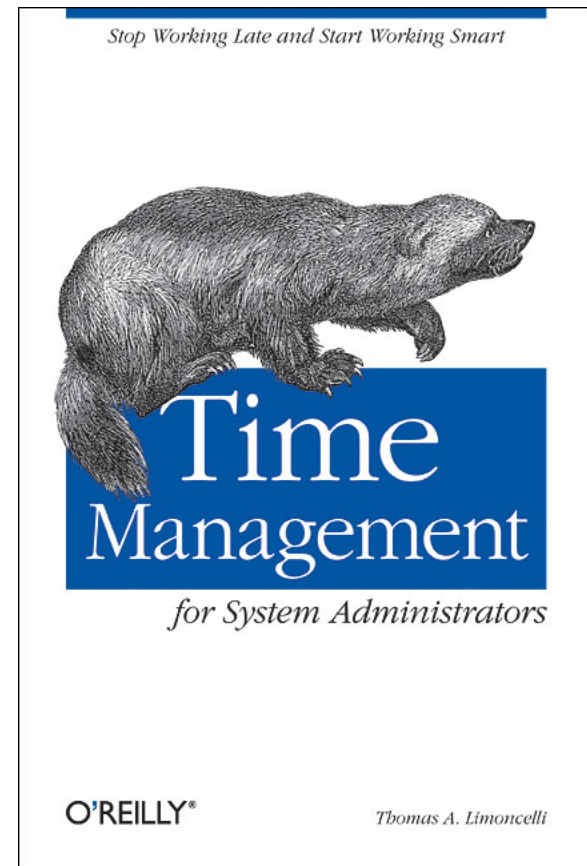


Another Bonus Example!

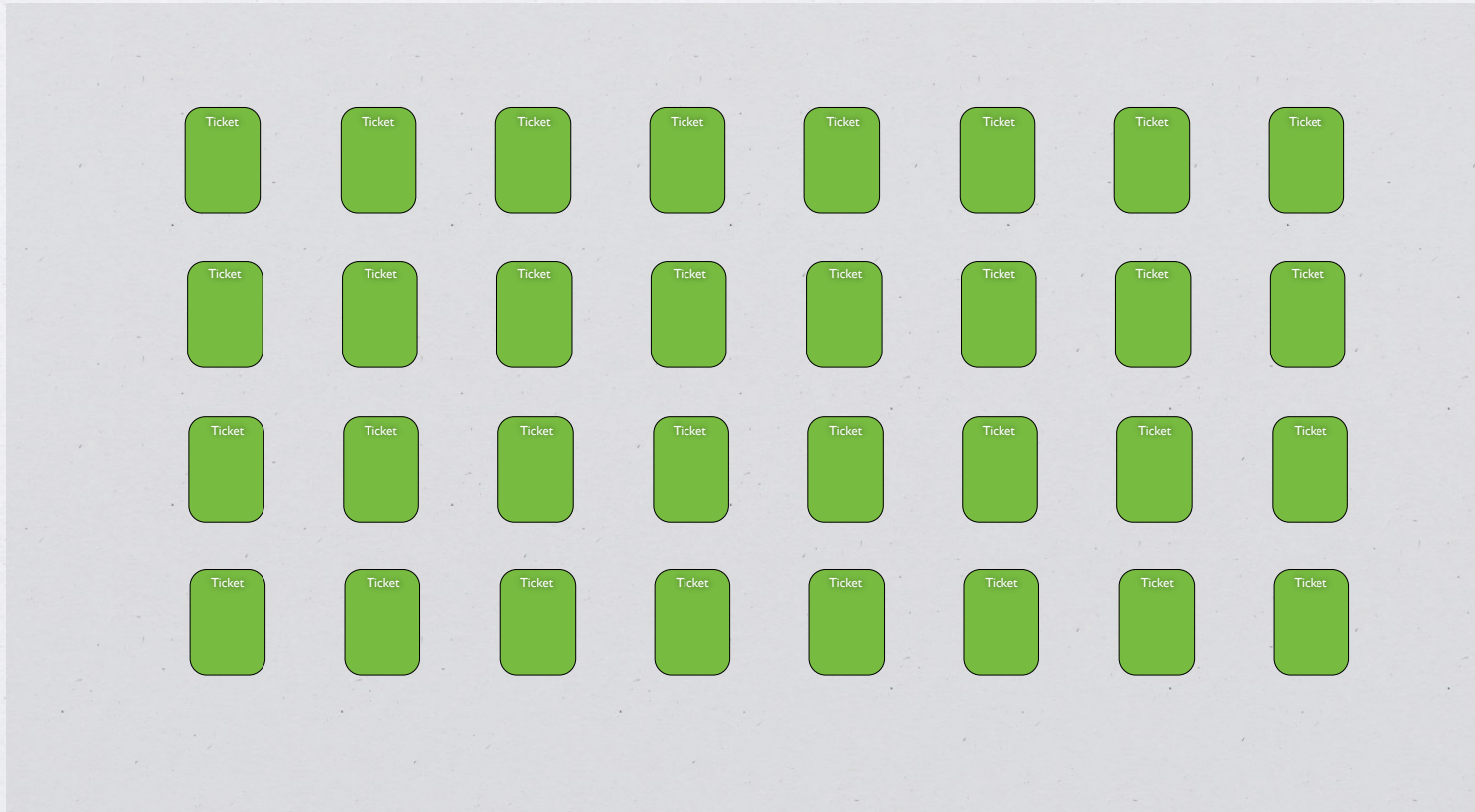
Personal Time Management

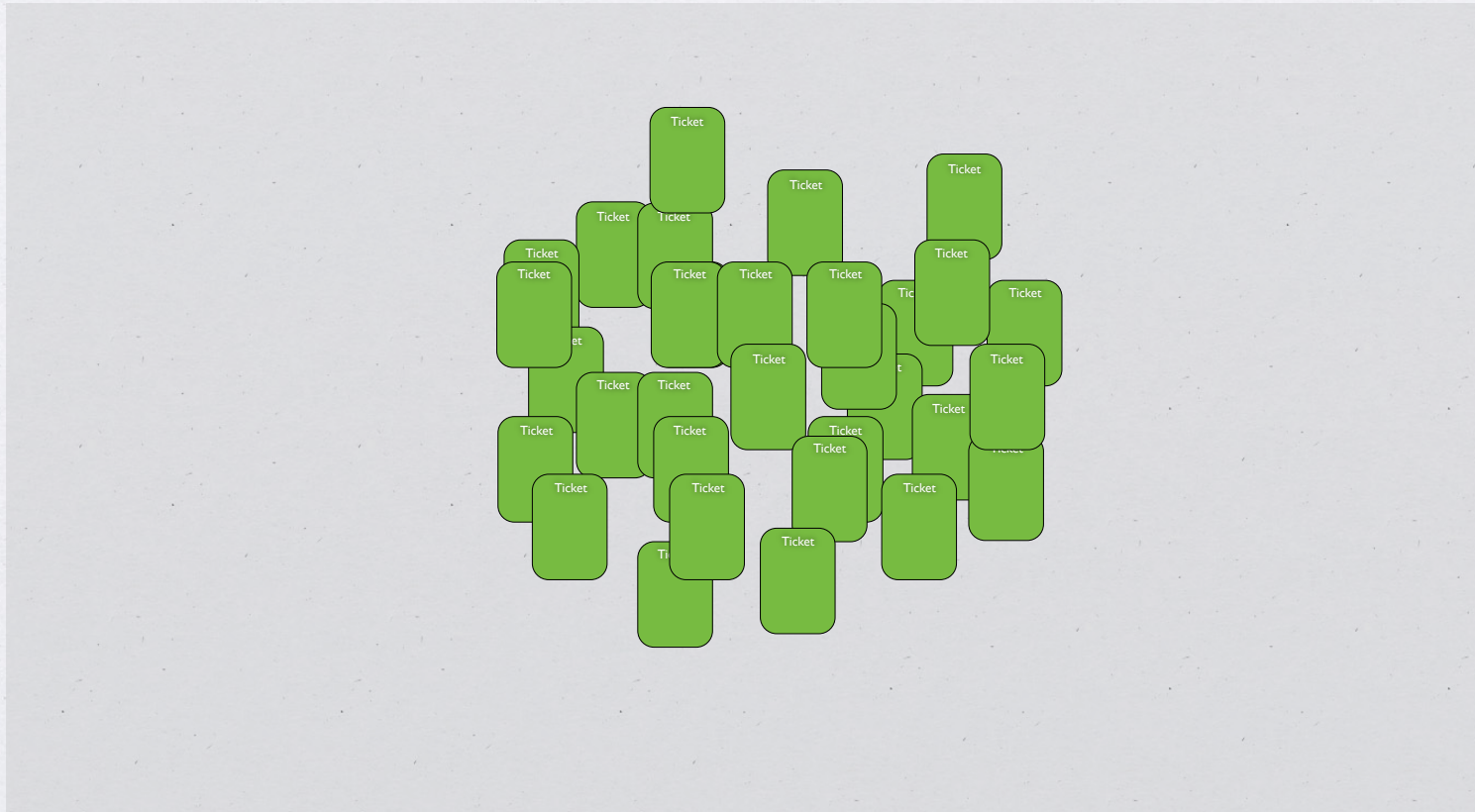
TM4SA “Start The Day Planning”

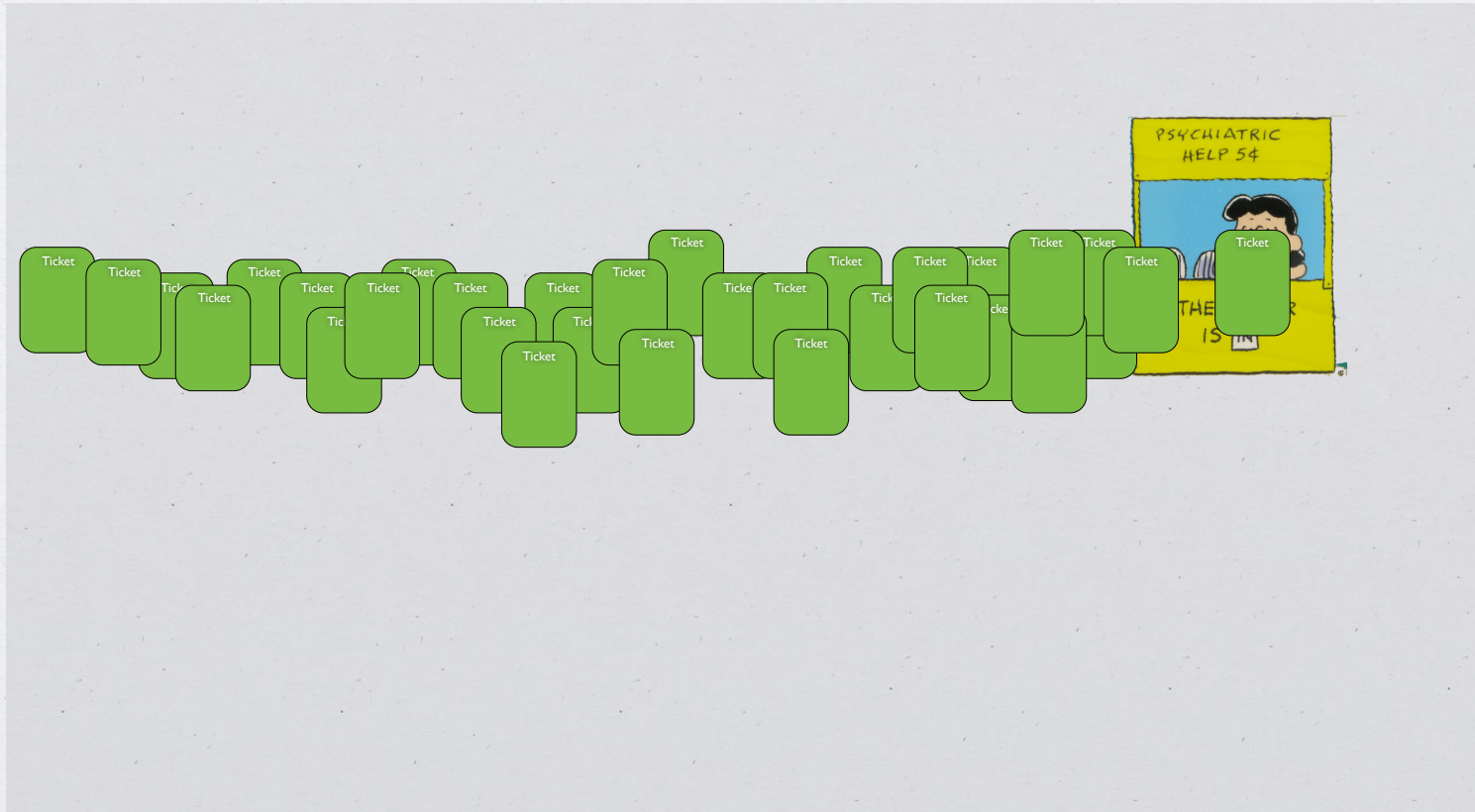
- Stop trying to do everything today.
- Take 8 hours of “work” from your todo list.
- Work on those items.
- Repeat every day.

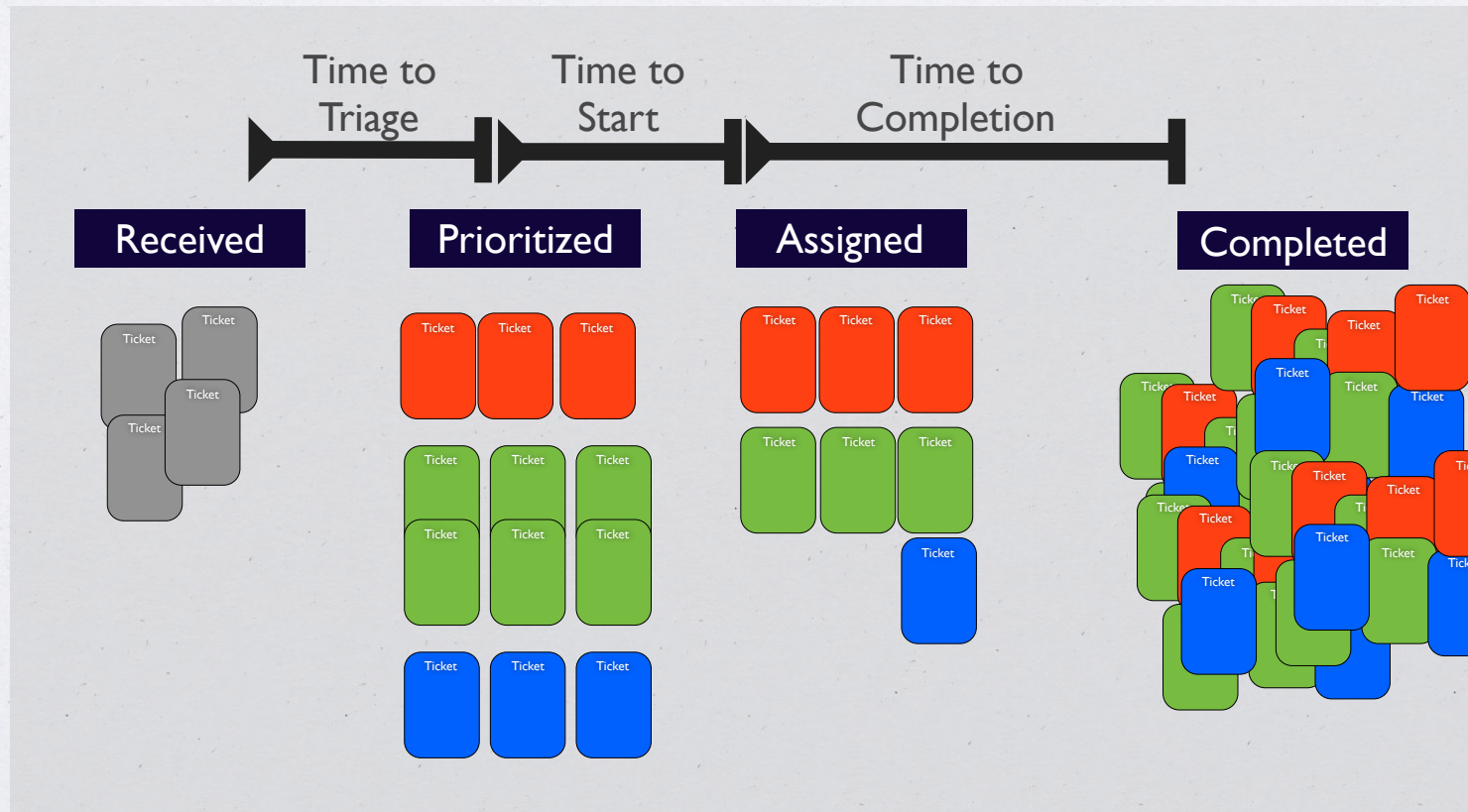


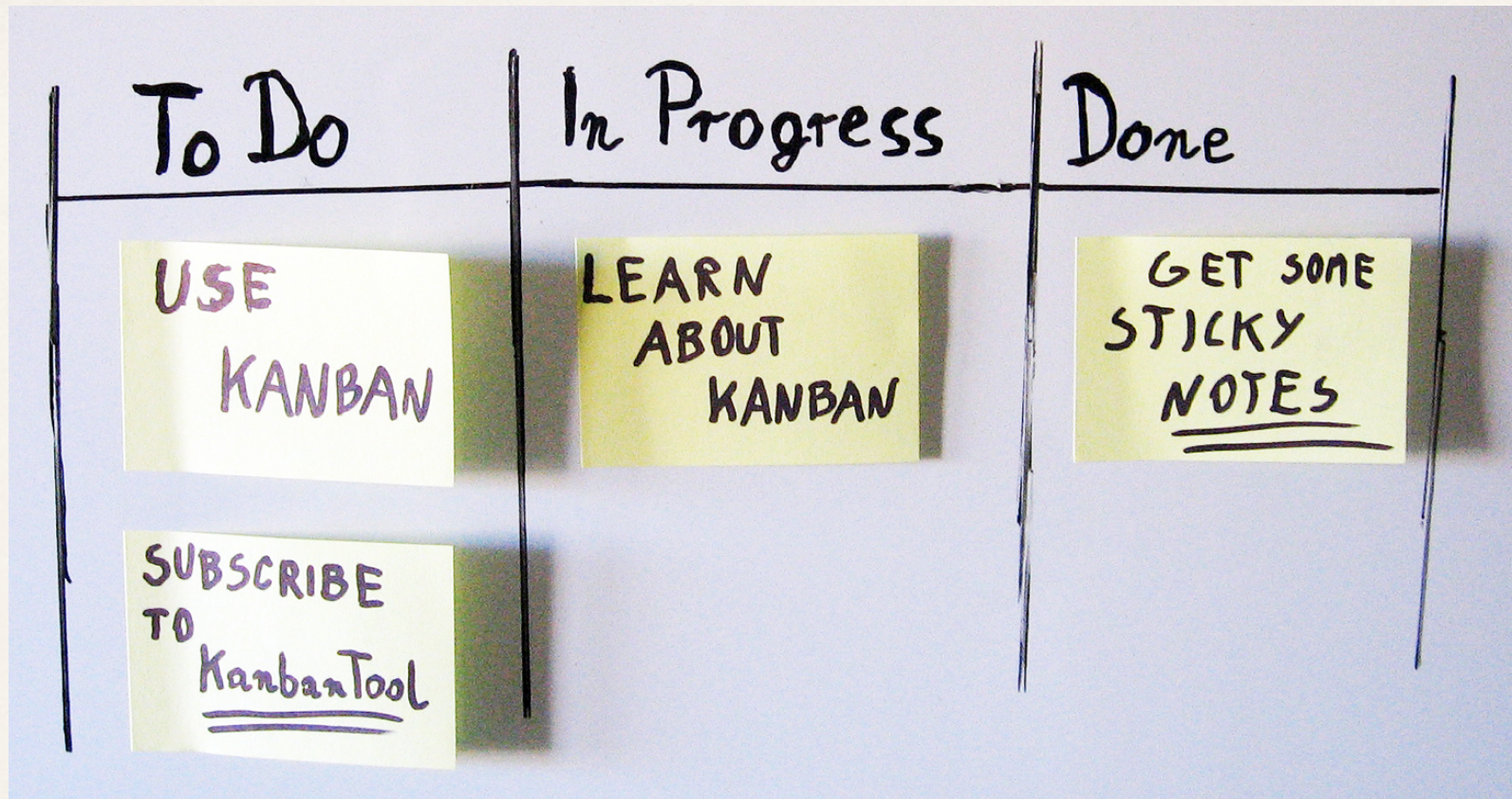
Kanban



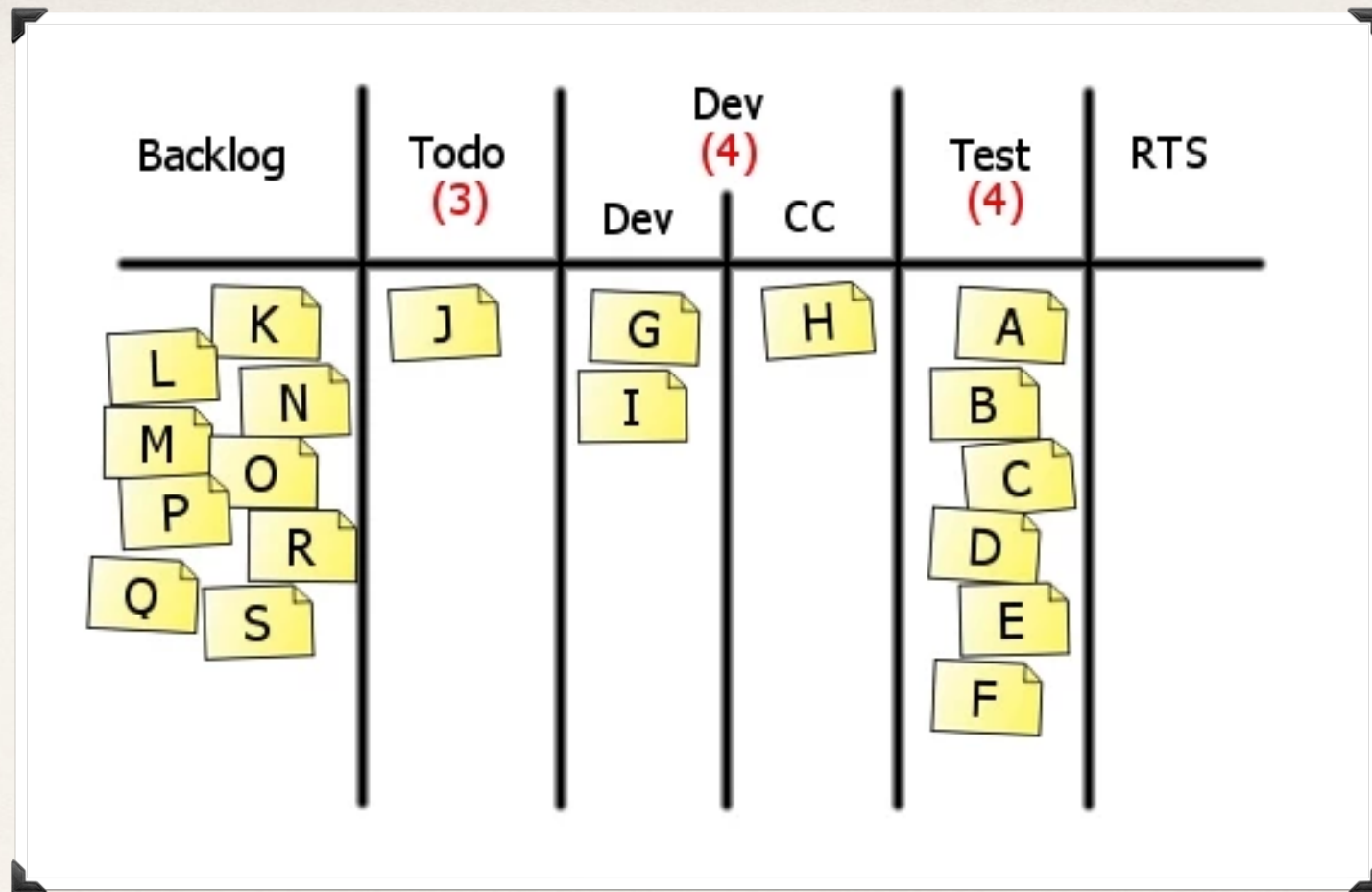




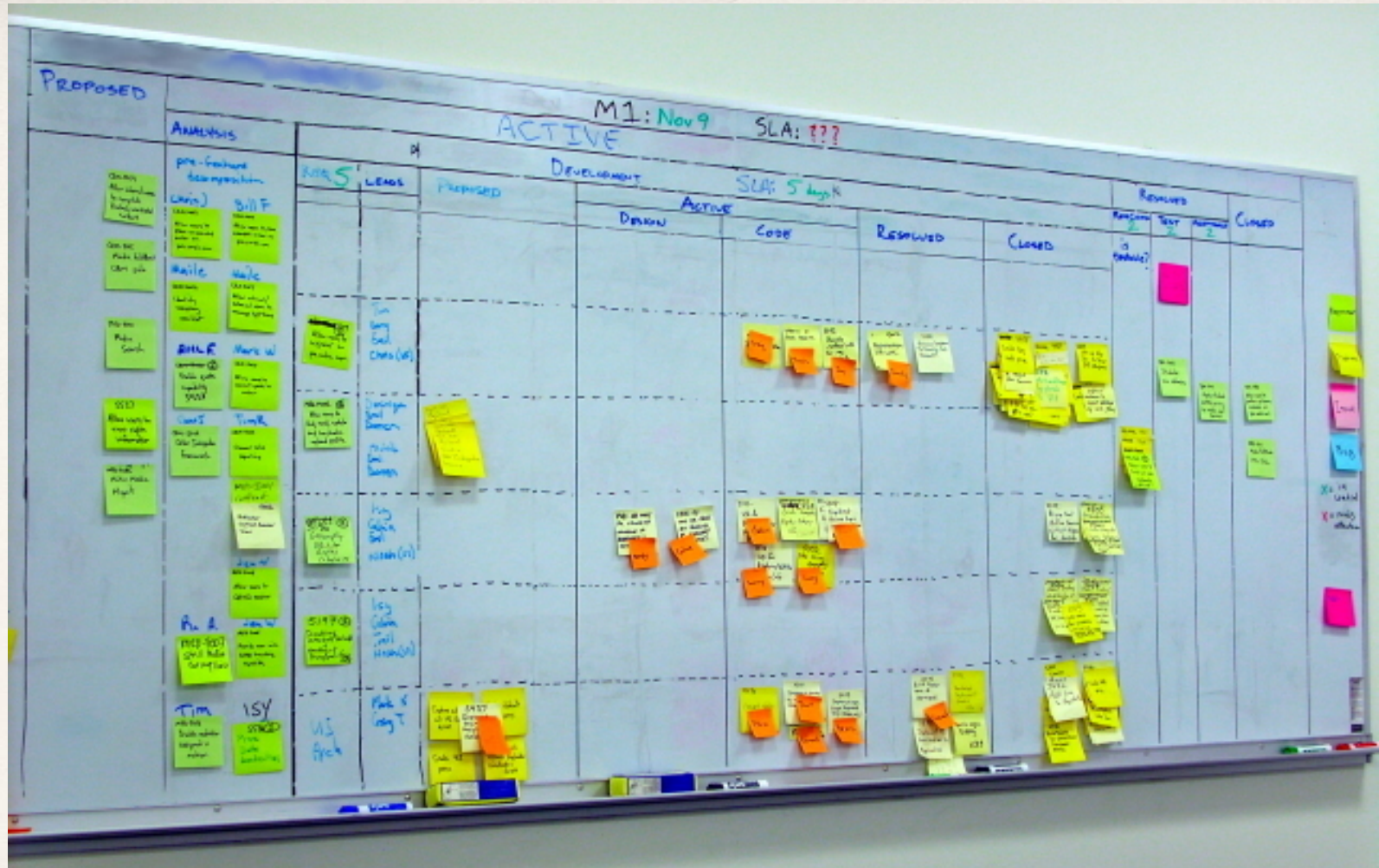




http://en.wikipedia.org/wiki/Kanban_board



<http://brodzinski.com/2009/11/kanban-story-kanban-board.html>



http://help.targetprocess.com/kanban_board/what

Kanban Books

- David Anderson's Kanban, Successful Evolutionary Change for Your Technology Business
- Henrik Kinberg's and Mattias Skarin's Kanban and Scrum - making the most of both
- Henrik Kniberg's Lean from the Trenches is great both at describing how to scale Kanban up but also as a practical guide to Kanban implementation
- John M. Gross and Kenneth R. McInnis's Kanban Made Simple

<http://pm.stackexchange.com/questions/3147>

Kanban Products

- Trello.com
- LeanKit.com
- (many others)

Kanban: The pull model

- 3 tasks per person in “active” each week.
- You “pull” three tasks through the system each week.
- If management adds more work, they can (1) hire more people, (2) reprioritize items.

Adapt to a helpdesk?

- A "fast lane" for quick requests.
- One person's "task for the week" is to handle all "fast requests" leaving the rest of the team to work on projects.
- Everyone's "3 per week" leaves enough time for occasional emergencies, interruptions, escalations.

Summary

- Push -> Pull
- Transparent. Lets other teams see how they fit into your priorities
- Management wants more velocity? They have to invest in optimizations or people, not “yelling louder”.
- Exec. mgmt can view and see the effects of their decisions

Bonus Example!

Apple Genius Bar-style help

User Help By Appointment

- Fixed number of appointments per day.
- Fixed amount of time per appointment.
- 1 issue or many
- “List all my annoyances”
- Many issues fixed in one setting.
- (emotional support too)

One of the core DevOps principles
is to go from "push" to "pull"

- Did I trick you into attending a DevOps talk? #sorrynotsorry

Take-homes

1. “Pull” is less stressful, healthier, avoidable.
 2. Work needs to be tracked, but it doesn’t have to push you.
 3. Reorganize:
 1. Assign SAs to projects. Don’t make projects push requests to you.
 2. Think in terms of “self service” rather than ticket/requests.
 3. Move to Kanban for individual projects, or ticket system.
- Help me! I need success stories to write about for my blog/books/etc.
 - Have a success story or case study? @YesThatTom

Homework

- Re-imagine your team task handling
- Adopt Kanban so that “pull” is used
 - Read “The Phoenix Project” by Gene Kim
- Adopt SRE ‘self-service’ model:
“Push the automation, not the people”

Transactional System Administration is Killing Us and Must be Stopped

Thomas A. Limoncelli, SRE

Stack Overflow

@YesThatTom EverythingSysadmin.com

usenix

LISA15

November 8–13, 2015 | Washington, D.C.

www.usenix.org/lisa15

#lisa15