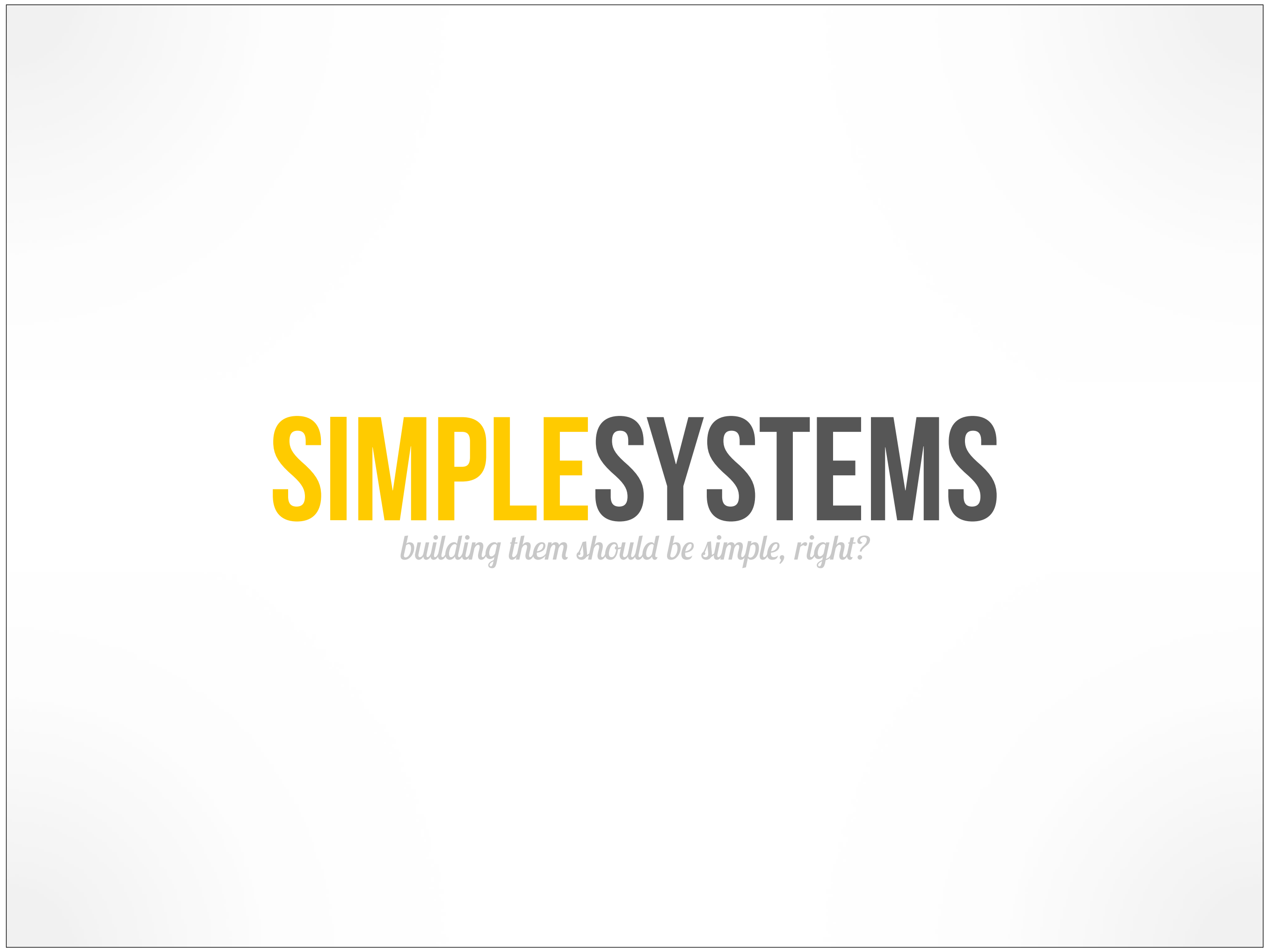
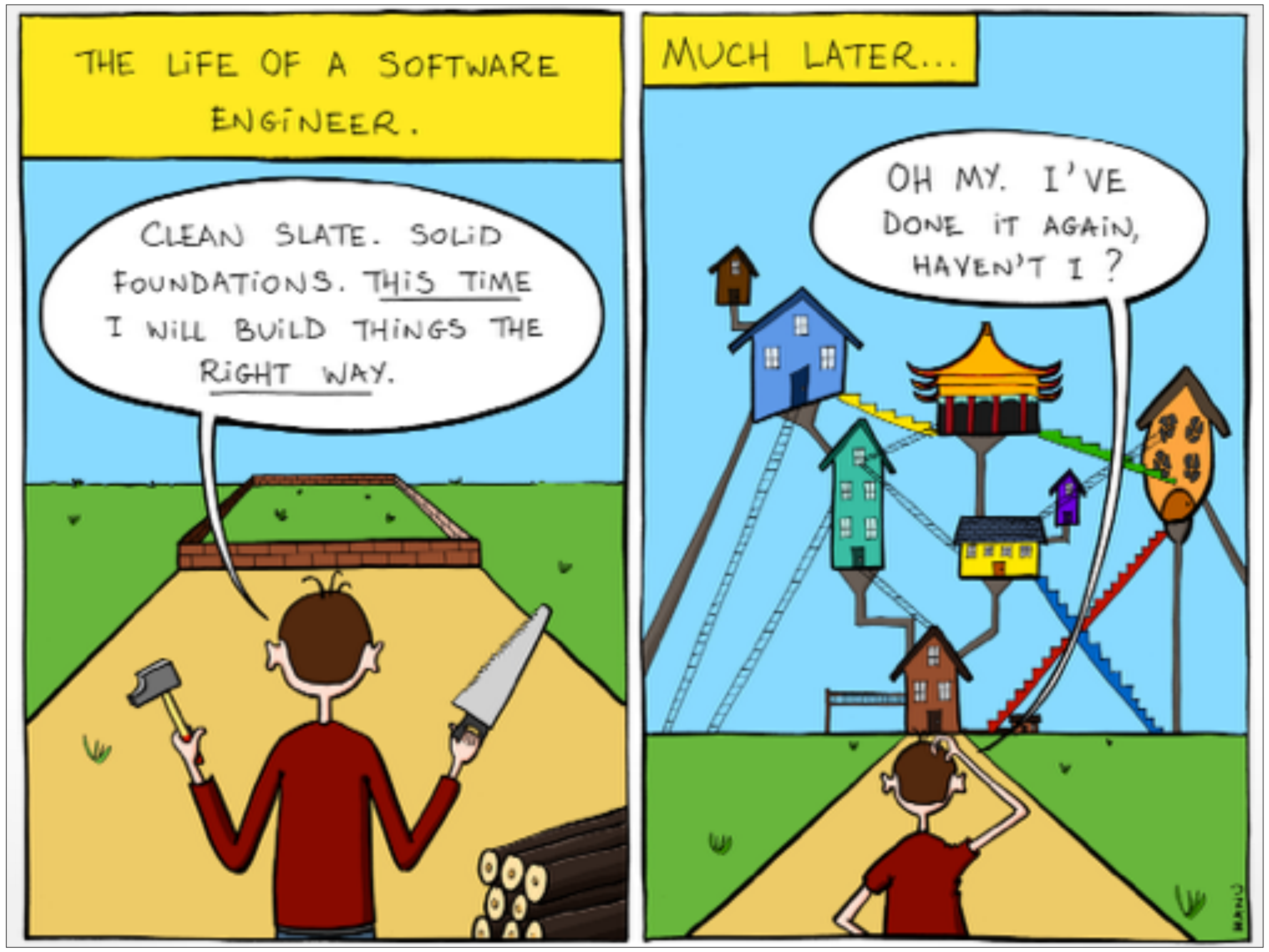


SIMPLESYSTEMS

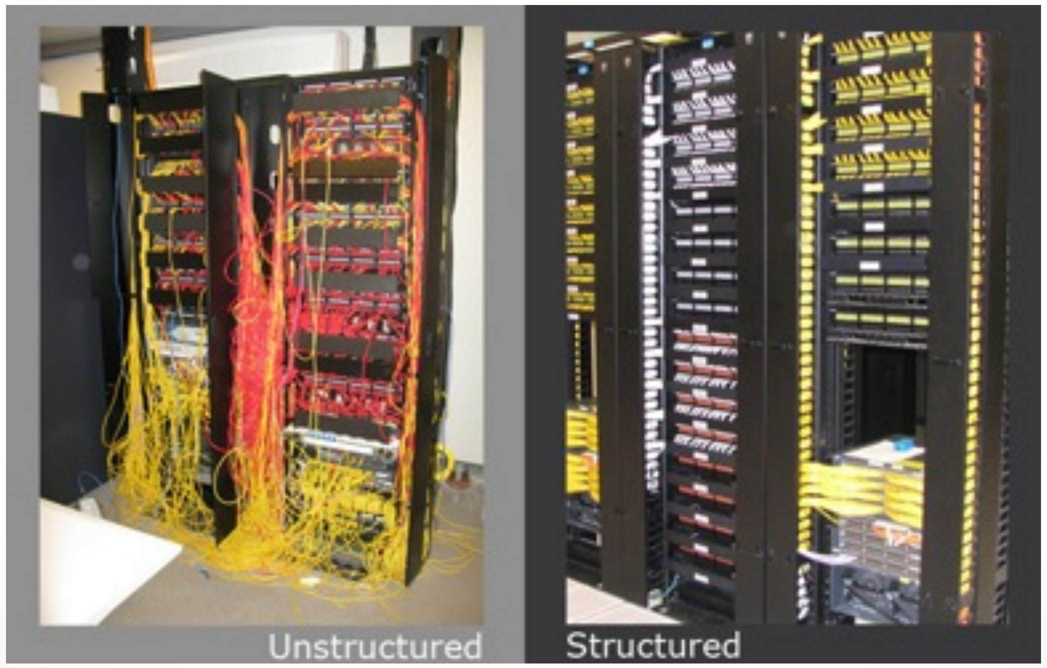
building them should be simple, right?

Simple systems should be simple to build
turns out they’re not.
What happened to me already a view times is the following


THE LIFE OF A SOFTWARE ENGINEER.
CLEAN SLATE. SOLID FOUNDATIONS. THIS TIME I WILL BUILD THINGS THE RIGHT WAY.
MUCH LATER...
OH MY. I'VE DONE IT AGAIN, HAVEN'T I ?

Turns out its not so simple to build simple looking systems.

COMPLEX VS SIMPLE



What do I mean by complex, monolithic, bloated, highly coupled

Complex is not always bad

why I hate

COMPLEXSYSTEMS

OFTEN HIGHLY COUPLED

THEY ARE ERROR PRONE

DIFFICULT TO MAINTAIN

HARD TO GET STARTED

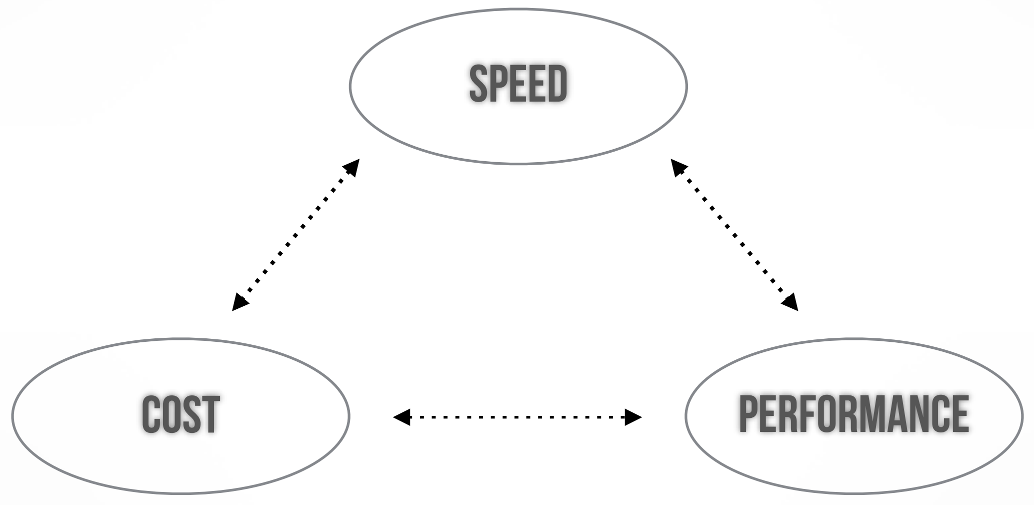
SLOW TESTS

LOTS OF DOCUMENTATION NECESSARY

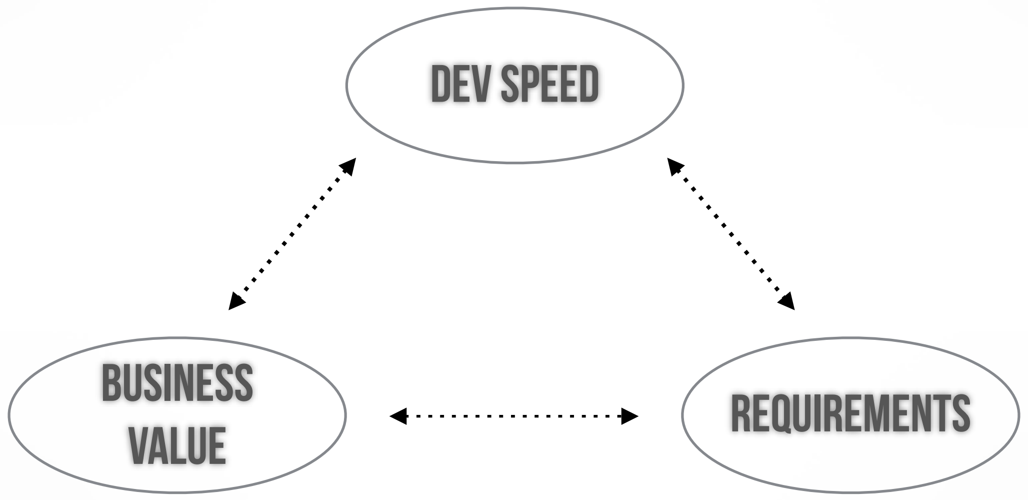
HARD TO DEPLOY

BUTWHY

REALITY IS HARD



REALITYISHARD



SO MANY OPTIONS

TECHNOLOGY X IS FASTER

BUT I LIKE FRAMEWORK Y BETTER...

OTHER PEOPLE HAVE TO USE YOUR TOOLBOX SO CHOOSE WISELY

I'M USED TO Z ALREADY

CONVENIENCE

QUICK AND DIRTY

POOR PLANNING

ITS JUST USED IN ONE PLACE SO…

DIVERGING FROM RACING LINE IS INEFFICIENT

I DIDN'T BUILD THIS / NOT MY JOB

For people its hard to let go of things

BUSINESSVALUE

PERCEIVED BUSINESS VALUE IN CLEANING UP CODE IS LOW

CHAOS SEEMS HARDWORKING

NEW FEATURES GET HIGHER PRIORITY

REFACTORING FOCUSES ON FANCY STUFF

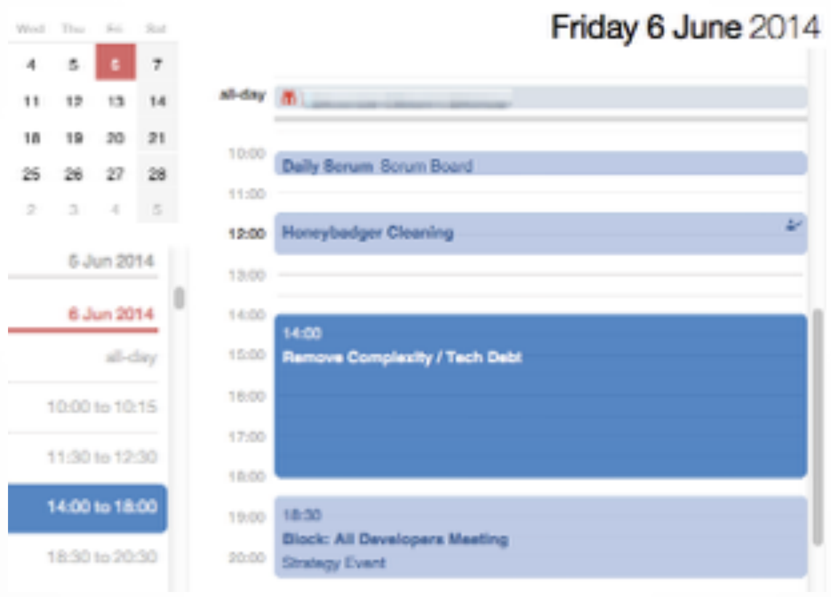
- Think two teams doing the same job. One of them spends lots of late hours fixing uber urgent problems. Lots of people sitting behind one PC
- Cross reference Hardworking vs Smart working

HOW TO IMPROVE?

Don’t do it, don’t build complex systems - not helpful

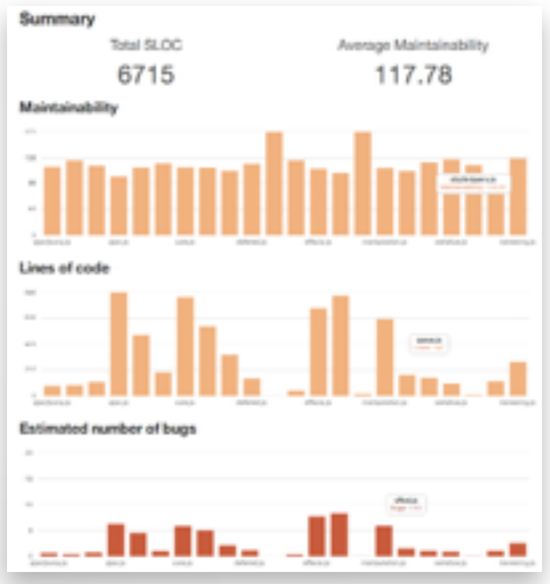
Won’t solve all problems but I’ll try to give some actionable hints how to improve.

DEDICATED TIMEBOX


Friday 6 June 2014
all-day
10:00 Daily Scrum Scrum Board
12:00 Honeybadger Cleaning
14:00 Remove Complexity / Tech Debt
18:30 Block: All Developers Meeting Strategy Event
5 Jun 2014
6 Jun 2014
10:00 to 10:15
11:30 to 12:30
14:00 to 18:00
18:30 to 20:30

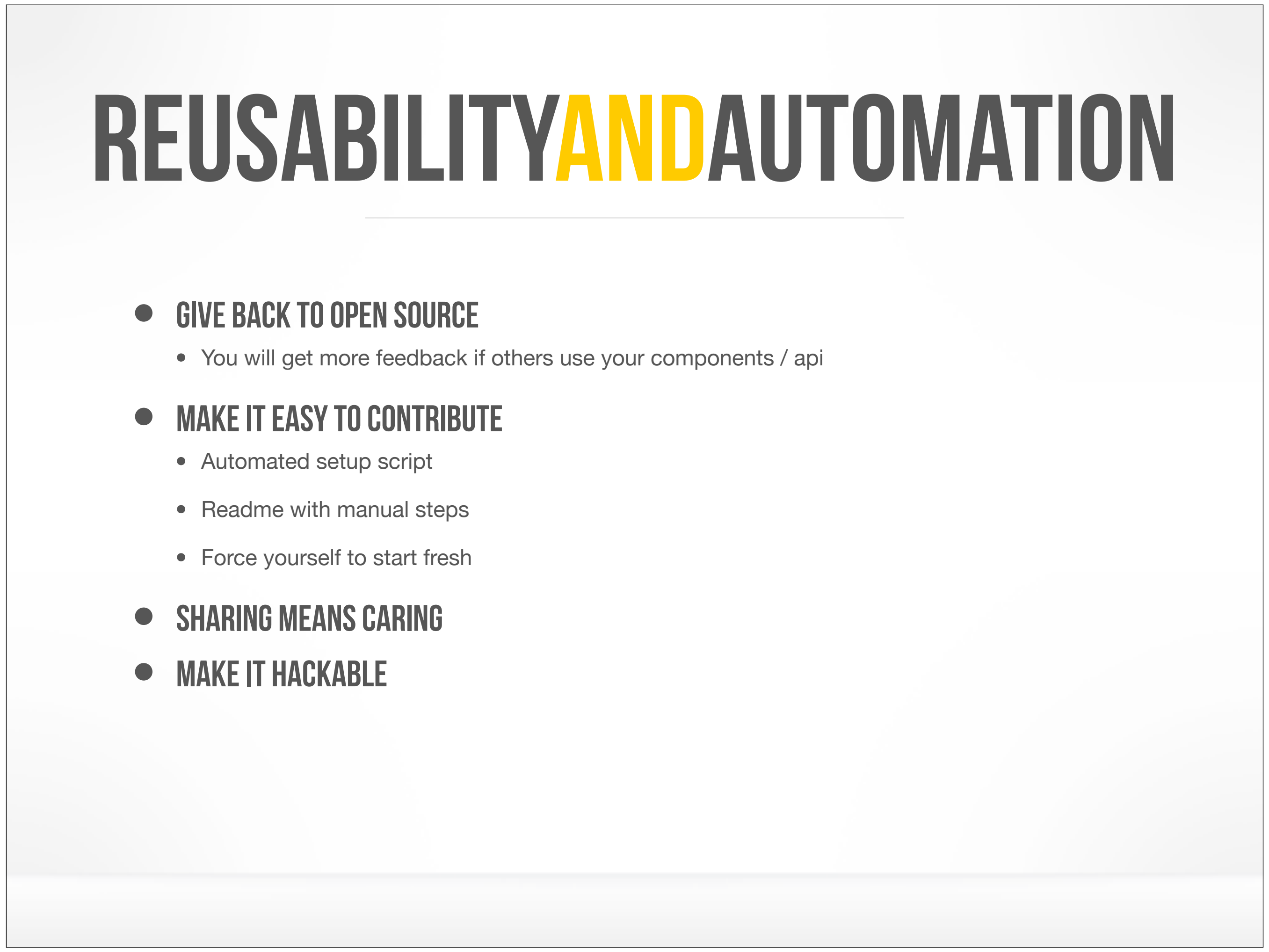
Consider Architecture Forum / Meetings
Spend time aligning you team
Also consider Complexity as technical debt
This also helps selling it

DATADRIVEN



- GAMIFICATION
- MAKE COMPLEXITY VISIBLE
- INCENTIVISE SIMPLE SYSTEMS
- GATHER DATA ON HOW HARD IT WAS TO ADD A FEATURE
- HOW HARD TO REMOVE

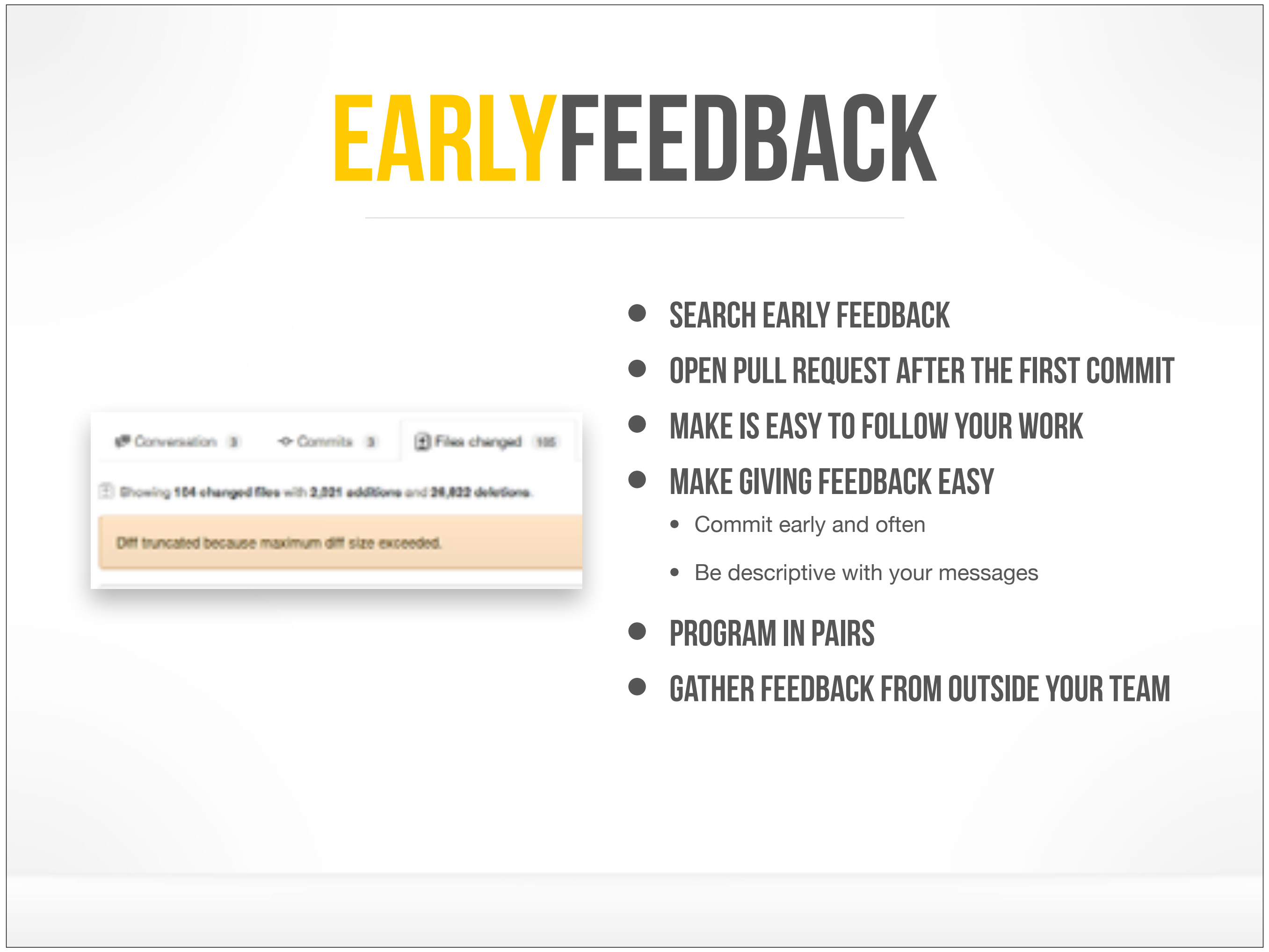
Make it visible what your concerned about



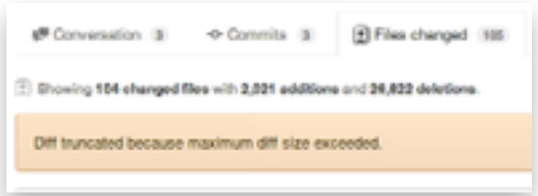
REUSABILITYANDAUTOMATION

- **GIVE BACK TO OPEN SOURCE**
  - You will get more feedback if others use your components / api
- **MAKE IT EASY TO CONTRIBUTE**
  - Automated setup script
  - Readme with manual steps
  - Force yourself to start fresh
- **SHARING MEANS CARING**
- **MAKE IT HACKABLE**

Open sourcing components will make sure they are self contained / clear interface / improve docs etc.



EARLYFEEDBACK



Conversation 3 Commits 3 Files changed 105

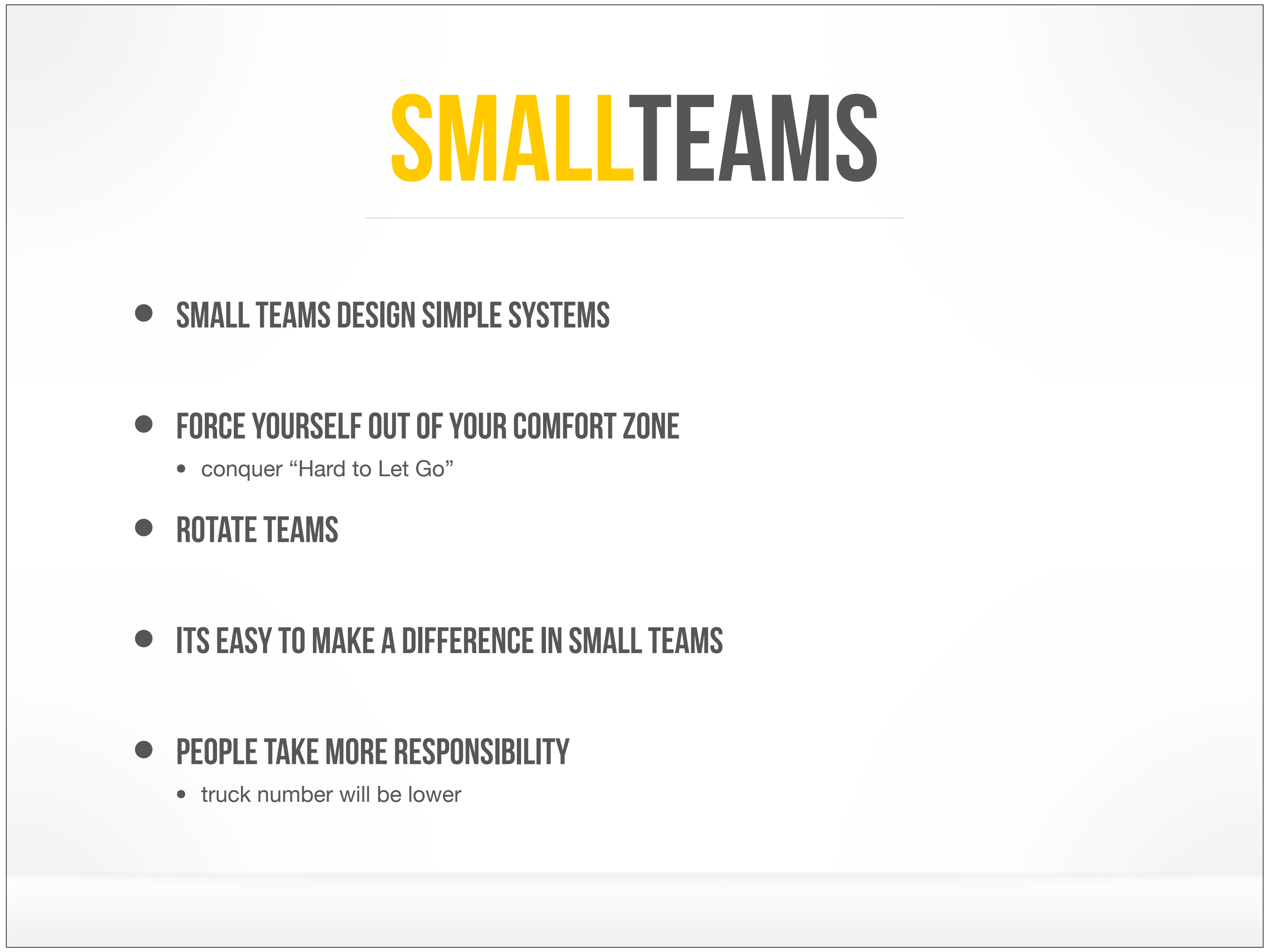
Showing 104 changed files with 2,021 additions and 26,822 deletions.

Diff truncated because maximum diff size exceeded.

- SEARCH EARLY FEEDBACK
- OPEN PULL REQUEST AFTER THE FIRST COMMIT
- MAKE IS EASY TO FOLLOW YOUR WORK
- MAKE GIVING FEEDBACK EASY
  - Commit early and often
  - Be descriptive with your messages
- PROGRAM IN PAIRS
- GATHER FEEDBACK FROM OUTSIDE YOUR TEAM

explain 30 % Feedback http://blog.42floors.com/thirty-percent-feedback/

most time is spend reading code -> make it simple to do so



SMALLTEAMS

- SMALL TEAMS DESIGN SIMPLE SYSTEMS
- FORCE YOURSELF OUT OF YOUR COMFORT ZONE
  - conquer “Hard to Let Go”
- ROTATE TEAMS
- ITS EASY TO MAKE A DIFFERENCE IN SMALL TEAMS
- PEOPLE TAKE MORE RESPONSIBILITY
  - truck number will be lower

- Helps conquer “hard to let go” problem
- Mythtical man month told us already if you have 3 teams you get a 3 layered system
- Possible problems:

* truck number lower but shouldn’t be a problem when rotating more often and products are small&simple

HARALDWARTIG

TWITTER.COM/HWARTIG

GITHUB.COM/HWARTIG

LINKEDIN.COM/IN/HWARTIG

FACEBOOK.COM/HWARTIG