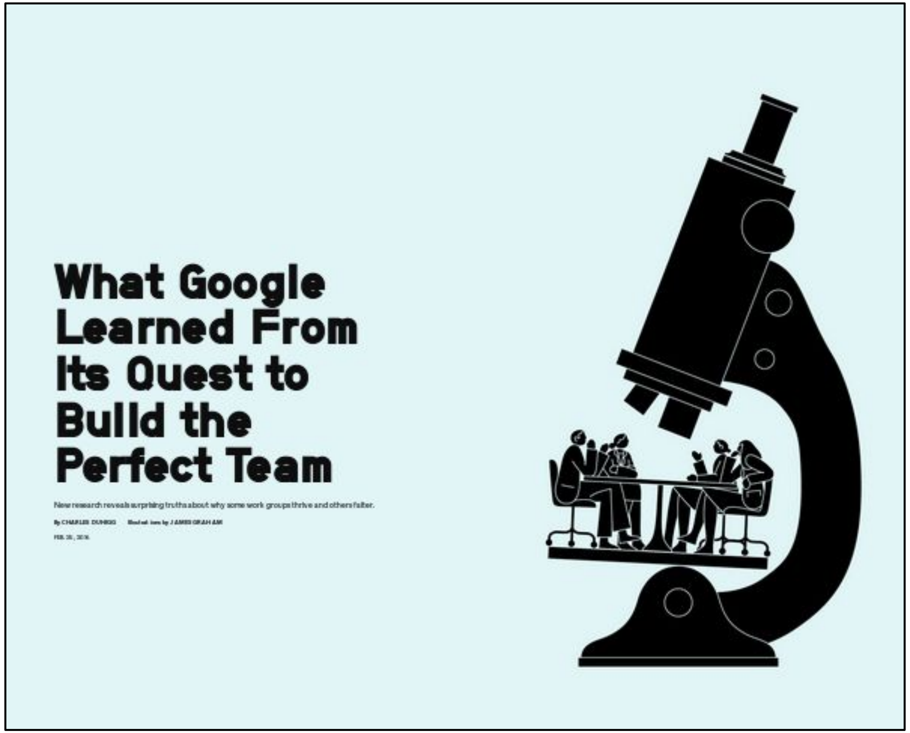
Git Workflow for Case Studies

Objectives

1. Congratulations on sticking with it (really).
2. What team characteristics lead to success?
3. How do you collaborate, as a team, with Git and Github?

What teams succeed



What Google Learned From Its Quest to Build the Perfect Team

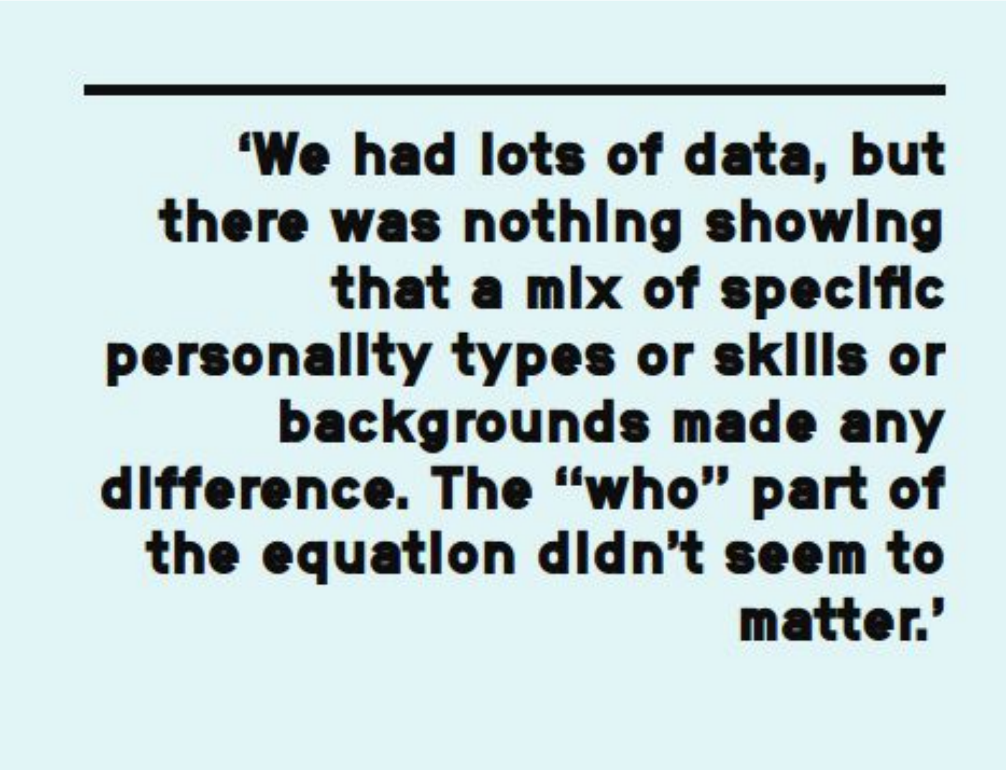
New research reveals surprising truths about why some work groups thrive and others falter.

By CHARLES DUHIGG Illustrations by JAMES GRAHAM

FEB. 25, 2016

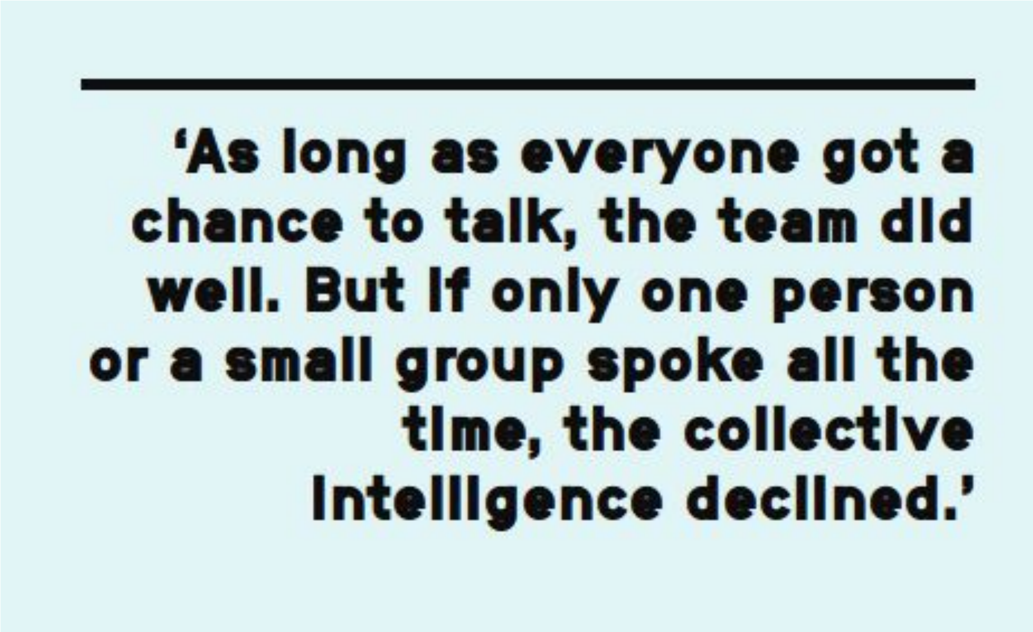
https://www.nytimes.com/2016/02/28/magazine/what-google-learned-from-its-quest-to-build-the-perfect-team.html

“Who” didn’t seem to matter



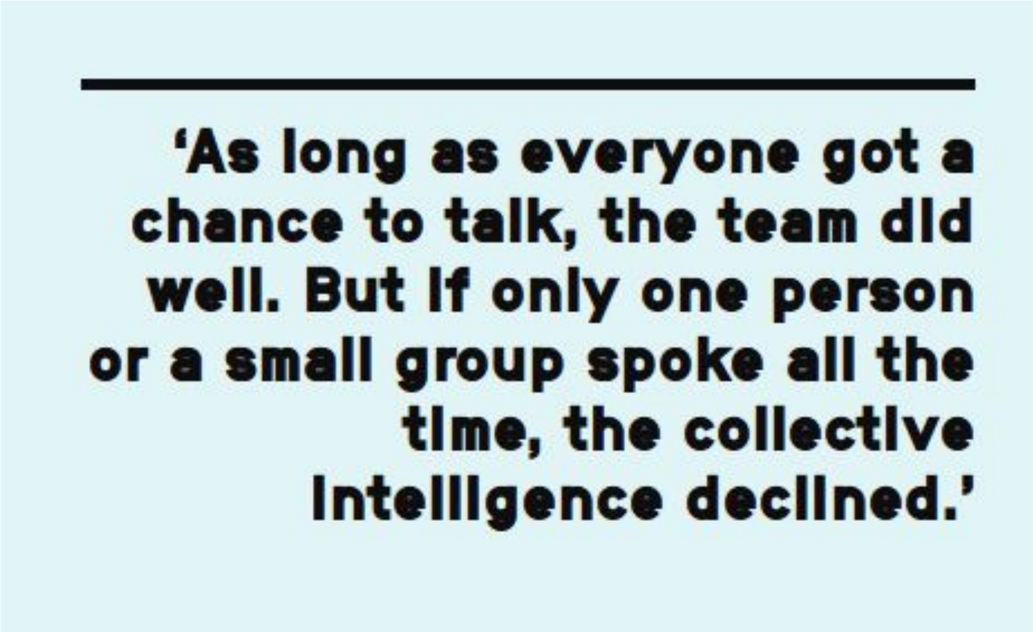
‘We had lots of data, but there was nothing showing that a mix of specific personality types or skills or backgrounds made any difference. The “who” part of the equation didn’t seem to matter.’

Goal: “Collective intelligence”



‘As long as everyone got a chance to talk, the team did well. But if only one person or a small group spoke all the time, the collective intelligence declined.’

Goal: “Collective intelligence”



‘As long as everyone got a chance to talk, the team did well. But if only one person or a small group spoke all the time, the collective intelligence declined.’

Don’t talk all the time. Listen to each other.
You’ll learn something, your team will likely do better, and you’ll like each other more.

Collaborating on Github on a group project

Goal: Team members will work on their own parts of the project but contribute to a Github repo that contains everyone’s contributions to the project.

Git Workflow Options (1 & 2 of 4)

- Centralized workflow (not recommended)
  - One person in group clones the gschool case study repo, adds other members as collaborators.
  - Everyone else clones that person's repo.
  - Everyone works in the master branch!

- Feature Branch workflow (recommended)
  - Same as above, but everyone works in their own branches
    - master branch is **not** for development - for production/working code.
    - name branches using group member names (e.g. fred, mary, juan, etc.)
    - as work is completed (in bite-size chunks), branches are pushed to Github and pull requests are issued on GH to merge the branch into master.
    - everyone is constantly making branches off master, doing work, then issuing pull requests on Github to merge their changes into master.

Git Workflow Options (3 & 4 of 4)

- Git Workflow
    - Similar to Feature Branch, but with more strict branch rules

- Forking Workflow
    - One person clones the gschool case study repo (call that repo *upstream*), everyone else **forks** the *upstream* repo and then clones their own repo locally.
    - Similar to how we do the pair assignments.
    - Each collaborator still makes and works in their own branches, but push their branches to their personal GH, and then issue pull request from their GH to merge changes into *upstream* GH repo.
    - Common for open-source projects.

References

Atlassian Git Workflows:
https://www.atlassian.com/git/tutorials/comparing-workflows

How to collaborate on git:
https://code.tutsplus.com/tutorials/how-to-collaborate-on-github--net-34267

Resolving merge conflicts using the command line:
https://help.github.com/articles/resolving-a-merge-conflict-using-the-command-line/

Resolving merge conflicts on Github:
https://help.github.com/articles/resolving-a-merge-conflict-on-github/

The difference between origin and upstream on github:
http://stackoverflow.com/questions/9257533/what-is-the-difference-between-origin-and-upstream-on-github