UNRULY

Mob Programming

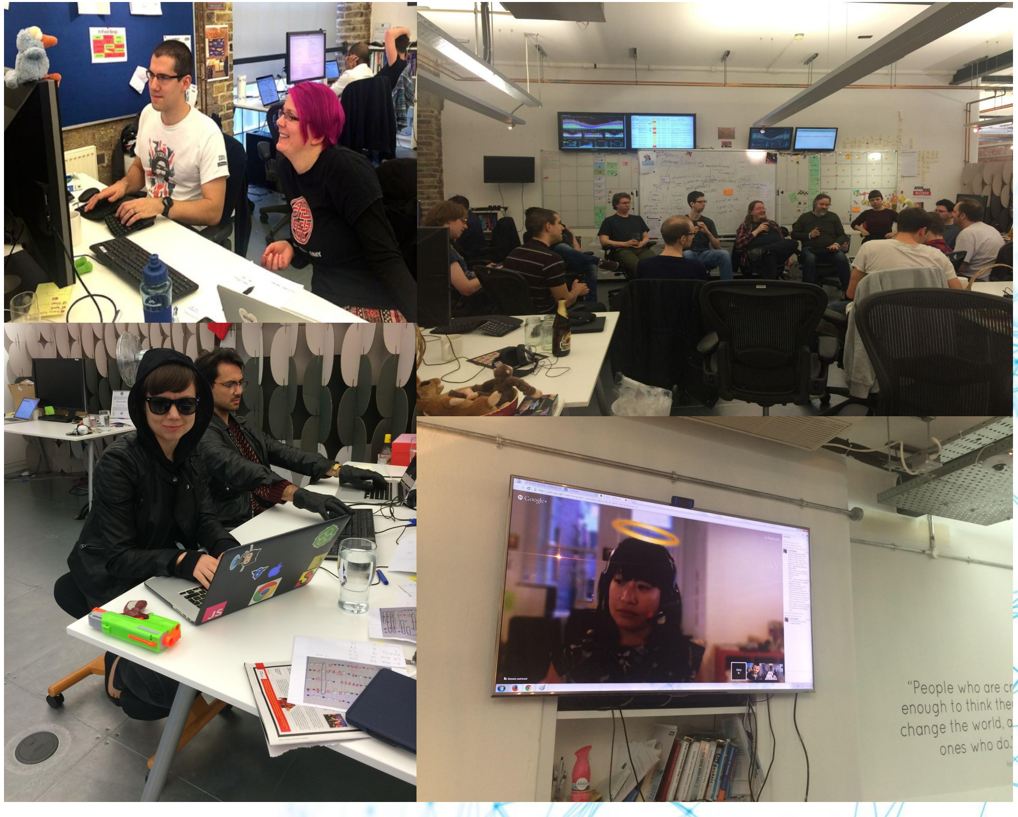
What Works, What Doesn’t

Alex Wilson
@pr0bablyfine

Who are Unruly?

UNRULY

- Advertising technology in London
- 200 employees globally
- ~25 Developers, 3 Teams
- Modern XP/Agile



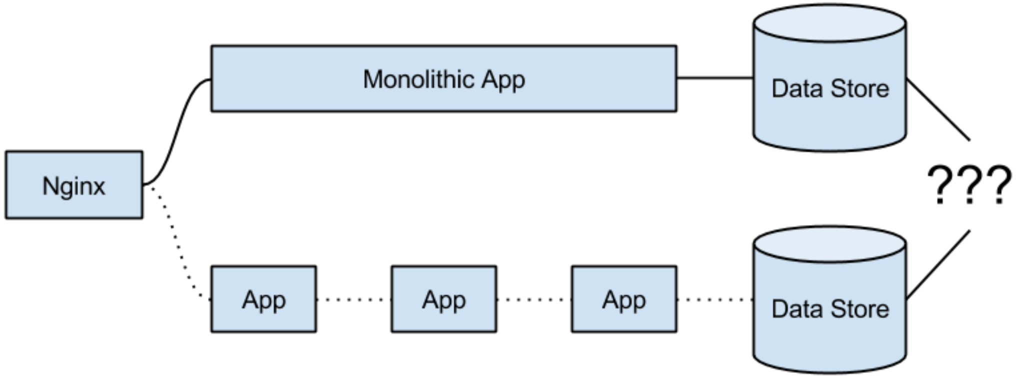
Unruly's Experiences

UNRULY

- Adopting Mob Programming
- Adapting Mob Programming
- Pitfalls and Dangers
- Questions

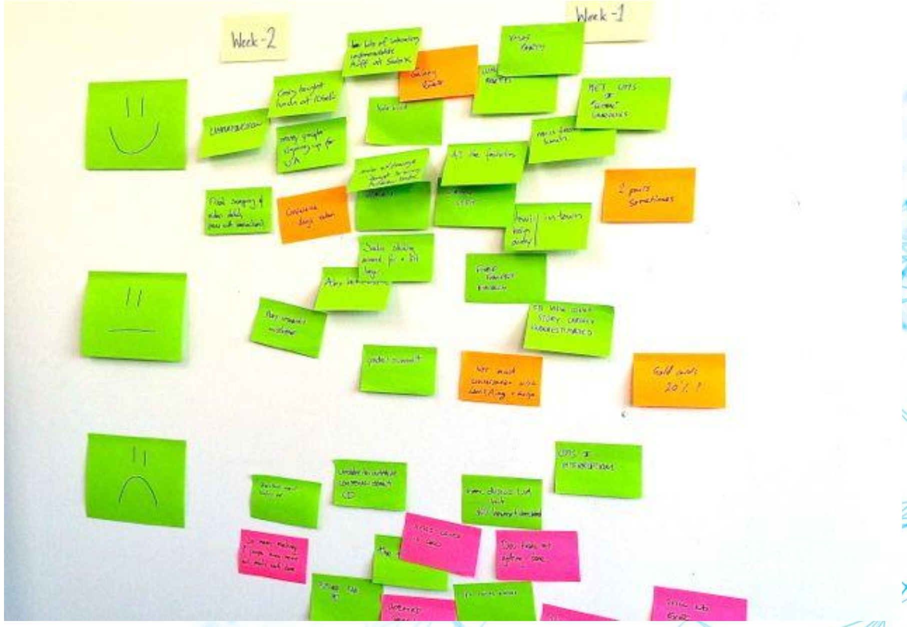
A Specific Problem

- Large-scale application migration
- Downtime and errors both translated to financial losses.
- “Verify” Branch by Abstraction



An Experimental Culture

- Developers are encouraged to embrace new techniques and technology
- We went to Woody Zuill’s talk on Mob Programming at JavaOne 2014



Adoption within Development

UNRULY

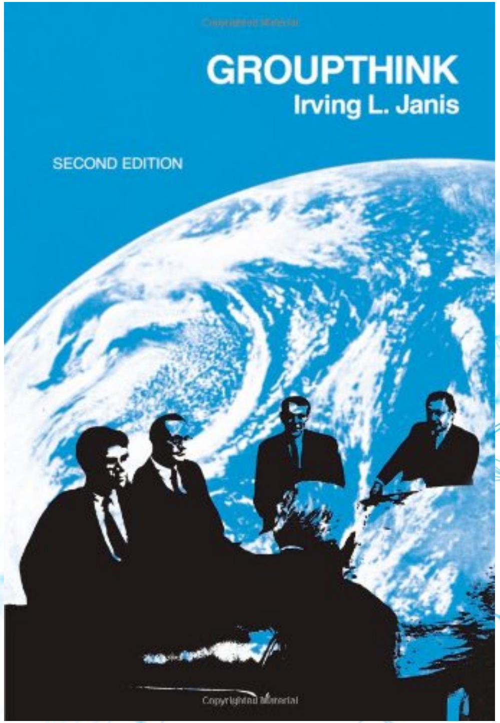
- We run an open-plan development area, separate from other departments

- Other teams began to adopt

- Team-specific variations to meet their needs

Problem: GroupThink

- Tightly-knit teams susceptible to GroupThink.
- Desire for harmony leads to irrational decisions.
- Team made a very odd decision early on, realised, and took action



Groupthink: Psychological Studies of Policy Decisions and Fiascoes

Amazon

Problem: Dominant Personas

- Team subconsciously gives more weight to “dominant” members.

- Not synonymous with loud.

- Nurture constructive dissent and include non-team members

Problem: Collective Narcissism

UNRULY

- “Together, the team can accomplish anything.”
- Risks seen as less of a problem
- Frequent team rotations highlight when the team is falling victim

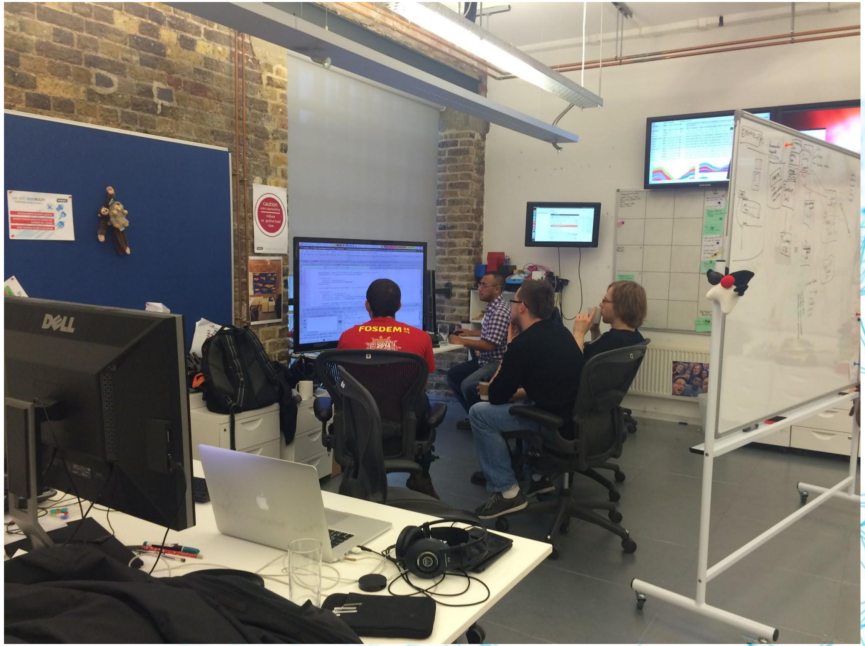
“Echo and Narcissus” by John William Waterhouse

Conclusions

- Mob Programming was useful for a specific problem.

- It’s fun! Doubles as a form of team-building.

- It will aggressively highlight existing dysfunctions in the team.



Thank you for listening.

Questions?

Alex Wilson
@pr0bablyfine