

# Questions I asked

- Tell me about your infrastructure
  - What is working well?
  - What has not worked well?
- What technologies are you considering adopting going forward?
- How do you capture and share lessons learned?

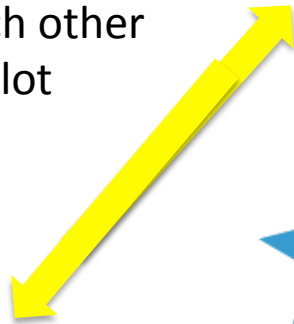
# Design challenge type of questions I asked

- Imagine that we tried to work together on a simple infrastructure project that say tried to coordinate / automate / trigger a network test on different infrastructures
  - The equivalent of astronomers asking each other to point their telescopes at interesting astronomical events
- How might we architect that? What technologies should we use?

# Accelerate this loop

## 1. Research Ideas / Questions

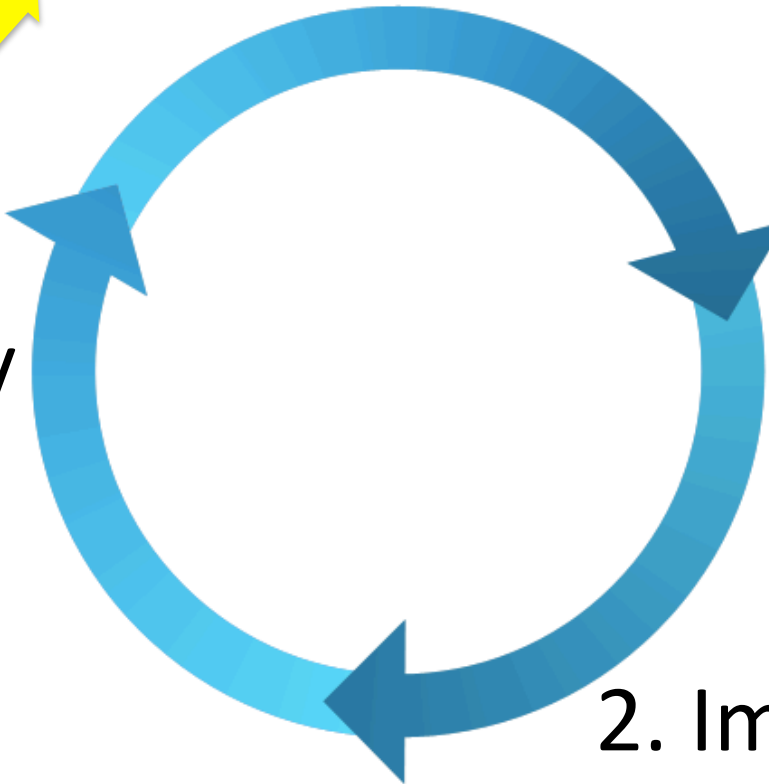
We talk to each other about these a lot



We should talk to each other about the engineering details more. It is fun, educational, therapeutic.



## 3. Evaluation / Results



## 2. Implementation

# Some high level takeaways

- Any piece of technology can work well at small scales
- Challenge is when you try to scale up
- Why not just get our best practices and lessons learned from top Google results?
  - We often shares workloads and challenges that are networking domain specific.

# Common challenges

Need to balance:

- Simplicity
- Scalability
- Performance
- Reliability
- Generality
- Features

As I was writing this slide I remembered a better list... so content here and next three slides are from:

<http://static.googleusercontent.com/media/research.google.com/en/us/people/jeff/stanford-295-talk.pdf>

# Software Engineering Advice from Building Large-Scale Distributed Systems

Jeff Dean

<http://labs.google.com/people/jeff>

# Design for Growth

**Try to anticipate how requirements will evolve**

keep likely features in mind as you design base system

**Ensure your design works if scale changes by 10X or 20X**

but the right solution for X often not optimal for 100X

# Get Advice Early!

- Before you write any code
- Before you write any lengthy design documents
  
- Instead:
  - Jot down some rough ideas (a few paragraphs)
  - Go find some people and chat at a whiteboard
    - Especially people familiar with building similar systems
  - Even better: discuss a few different potential designs & evaluate...



# Capturing infrastructure engineering howtos

- Value of non-authoritative engineering notes (if you can search efficiently)
- Not necessary to have one wiki that is the right answer (creates barriers to participation)

# Other takeaways

- Experience tracks
  - Don't just accept paper from Google / Yahoo / Facebook
  - Experience of academic community is valuable as it is going to be much closer to what you will experience
- Hybrid systems (build or buy)
  - Something many of us are evaluating