- Eliminate QA as a separate department, distribute them into the feature teams.

Turn Ops into the platform team and internal consultants, with these goals:
- provide devs the infrastructure they need
- compete with a vast army of experts who are there to help
- look for ways to make devs productive

(UP, 96%)

I want to be my company's spokeperson to technical audiences to advance the company brand and facilitate recruiting, competing with top tech companies for talent.

The Five Ideals:

1- Locality and Simplicity

2- Focus, Flow & Joy

3- Improvement of daily work

4- Psychological Safety

5- Customer Focus

(UP, 100%)

- Google's five keys to successful teams:
  1- Psychological safety: can teams take risks without feeling insecure?
  2- Dependability: Can teams count on one another?
  3- Structure & clarity: Clear goals, roles, execution plans?
  4- Meaning of work: Is the work important to the team?
  5- Impact of work: The team believes their work matters?

  (RTI p1)

- Team cognitive load assessment: (bit.ly/3U4yPUC)
  - How is the experience of building your services?
  - How is the experience of testing your services?
  - How is the experience of deploying your services?
  - How is the experience of operating your services?
  - How is the experience of being on call for your services?
  - How is the experience of dealing with industry regulations and compliance?

  (In the link above, will find suggestion of things to consider for each question)

- We should track dependencies between teams now and over time (RTI p5)

- Instead of spending time waiting on other teams to finish their work, focus on tracking then removing these in-flow dependencies. (RTI p5)

- Mik Kersten: "As long as a dev tool like Jira is not connected to a service desk tool [...] the corresponding lack of flow and feedback between these two systems creates an information bottleneck..

- In a remote-work world, it's essential to be clear all the time about:
  - What you're working on
  - Why you're working on it
  - How your work is being completed
  - When it should be completed by (RTI p5)

- Don't make it hard for people to discover meaning in written communications. Make messages self-contained. (RTI p6)

Help reduce cognitive load

Closing the Reliability Gap (bit.ly/3DY66el)

Where common reliability approaches fall short

Observability & Incident Response	Reactive by nature; doesn't prevent issues not yet seen
Chaos Engineering	Not measurable; hard to adopt and scale
SREs	Expensive & difficult to scale; Inconsistent standards & practices
SLOs & SLIs	Measures unreliability; doesn't indicate where to improve

- One of the goals of Team Topologies is to reduce dependencies between teams by increasing ownership of e2e systems/services to individual teams (RTI p~~[illegible]~~14)

- It is important to recognize that eradicating all dependencies is never 100% achievable. (RTI p~~[illegible]~~14)

- In practice, we should look to record the dependencies that a team has and identify:
  - Which ones are in control (for now)
  - Which ones are problematic
    - must be addressed
    - or must be removed (RTI p14)

- When tracking dependencies, the important thing to note is not the documentation in and of itself, but rather that the tracker can be used to assess how dependencies should change. (RTI p14)

- Dunbar number and military groupings:

People	Military name	Anthropological equivalent
8	Section	Hunting party
30-50	Troop	Tribe
100-150	Company	Village

↳ What's important is the fractal nature of these groupings: at each level, trust is maximized and there is no "us and them" within a grouping. (RTI p25)

Map online spaces to streams of change (or families of streams of change) (RTI p28)

↳ The seperate online spaces help remind people of the different boundaries within the architecture, and therefore boundaries within teams

↳ We can still have ~~[illegible]~~ shared channels between different online spaces to create porous boundaries (e.g. Slack Connect's shared channels)

- Lack of consistency about slack channel names, display names, the meanings of emoji, etiquette,
  - ↳ These things make the chat tool essential to watch (to avoid missing vital message) & confusing and difficult to use.
    - ↳ Instead, define a set of conventions that improve predictability and discoverability. (RTI p34)

↳ Increases cognitive load

- Having channel names to make it obvious where to get support (#support-...)
  - ↳ Makes it easy for people to "self-serve".
    - ↳ Especially if they have bookmarks in channel header to provide instructions. (RTI p35)

↳ Saves someone else's time

- We can use Slack Workflow builder templates to standardize the structure of requests coming into the team (RTI p35)
- Set conventions around the display name that shows in the chat for each person
  - ↳ "Jim" ⇒ "Jim Ngo (Infra platform team)"
  - ↳ "sara_b" ⇒ "Sara Brown (Green Stream Team)"
    - ↳ This is better than putting info in profile, because gives immediate context who that person is when we @ them.. (RTI p35)

Slack & Jira

- We need to create more purposeful interactions between teams.
  - ↳ The "broadcasting" approach to information sharing is never the best.
- High-trust organizations perform better

  ↓

  Achieved by providing consistent experiences for others in the organization. (TT p142)

  ↓ (and consistent mental models of our systems)

  This is because, humans work best with others when we can predict their behavior. ~~p142~~ (p24)

  ↓

  It's also why we hangout and have fun Gadgets (p87)

  This is where the Team API comes in..

  It creates more purposeful interactions between teams.

  We do this to leverage Conway's Law

- We can map team interactions happening between teams (e.g. bit.ly/3UrPdOO)
  - ↳ but remember this is just a snapshot. Should not become a model to follow mindlessly. (RTI p40)
- The problem with the organizational chart is that for a lot of knowledge work, we cannot have communication only happening up and down the different branches. (RTI p43)
  - ↳ Don't build communication channels around org chart.

- In chat tools, public channels should be created to enable people outside of a team to ask questions to <u>the team</u> rather than <u>to individuals</u> within the team.
  ↳ Help reduce occurences of people reaching out to individuals (RTI p47)

- Types of channels to have:
  - communities (areas of practice)
    - UX
    - testing
    - architecture
    - specific technologies and tools
  - Topic (long or short lived)
    - Trying to solve a specific issue
    - will typically be archived someday (RTI p47)

- Have a "public by default policy" to provide opportunity to learn from any discussion but ensure people's focus is narrowed to info useful to them. (RTI p47)

- In platform development, being transparent and engaging with customers of the platform in focused communication channels is a key contributor to success. (RTI p51)

- Use a net promoter score (NPS) to gauge how well the platform is perceived by internal customers. (RTI p57)

- The simplest user personas characterizes the goals & frustrations of each type of dev working with platform:
  - What do they want to achieve?
  - what do they find difficult? (RTI p57)

- Chat tool ecosystems typically have several options for automated onboarding help using bots.
  ↳ discover or predict which channels they should join (RTI p58)

- A guide to Slack, from Equal Experts: (bit.ly/3UT961B)

New Book!

Making Work Visible – Dominica DeGrandis

- Questions to check if you truly know your business:
  - What problems do we solve for our customers?
  - How do we specifically solve those problems?
  - How do we do it differently from our competition?
  - What is our proof? How do we validate our credibility?
  - How does our company's $ performance compare with the rest of our industry?
  - Why do our customers choose our product over other options? (MWV p xvi)

- Preparing for objections with key metrics
  ↳ bit.ly/3A7geiI (MWV p xviii)

  ↳ Which metric provide which insight (and solves which manager's complaint)

- Remove the term "best practice" from conversations about what to do.
  Recognize best practices are frequently usurped by better emerging practices. (MWV p xix)

- Busyness often means just doing so many things at once that they all turn out crappy. (MWV p xxiii)