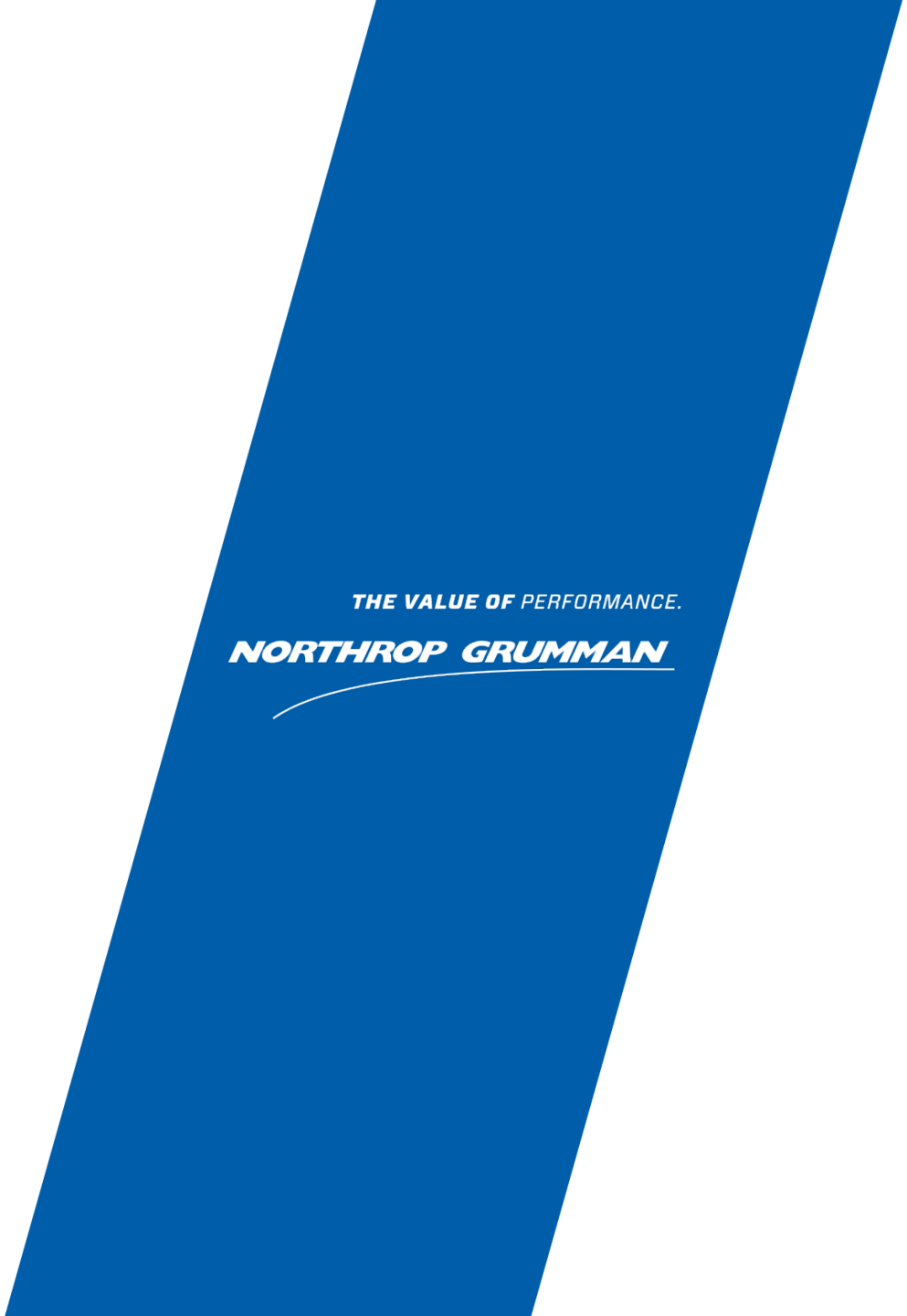


Refactoring Team Interaction

Dealing with Dysfunctions within the Team

7 June 2018

Scott Grimes

Software Engineer/Product Owner

GAP CIE

- **Global Adaptive Planning Collaborative Information Environment (GAP CIE)** is a Web-based, collaborative Joint Operation Planning (JOP) tool for the combatant commander and strategic level.

- Changed the way software Development happens on GAP CIE
  - Agile (Scrum) Software development practices
  - DevOps, continuous delivery practices
  - Internal “Learning Organization”

GAP CIE Mission Statement

- Foster an organizational culture that empowers cross-functional teams working across the value stream to deliver the highest quality products while acting with intention and adhering to the Agile Principles.

Intro

NORTHROP GRUMMAN

- Iowa State BS CS Grad, University Of Nebraska (Omaha) MS CS
- Software Dev (Java, C++, AngularJS, Perl, Python, Ansible, Docker)
- Large, multi-year projects
- Chief Product Owner: GAP CIE
- DoD Contracting
- CSM, CSPO, CSP, PSM-I, PMI-ACP
- Co-Organizer of Agile for Defense in Bellevue, NE.

Topics

- Demotivation
- Disempowerment
- Communication trouble
- The problem with estimation
- Lack of Responsibility/Leadership

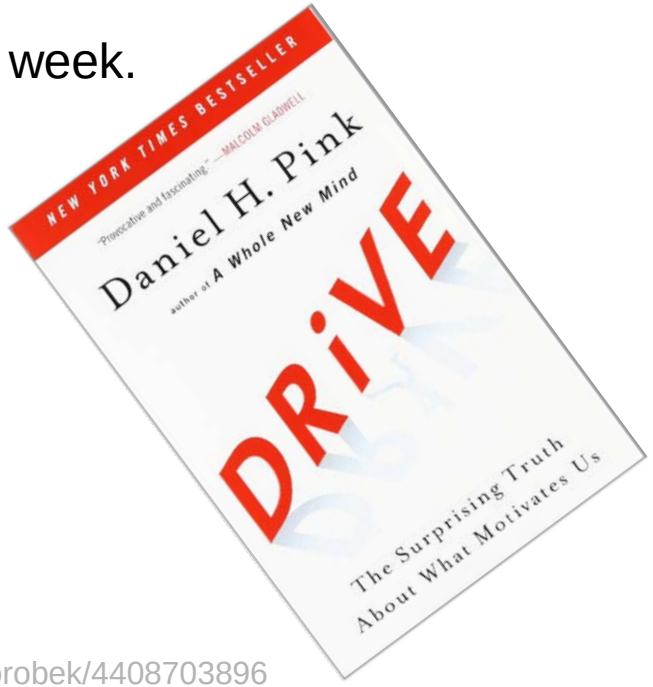
THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Demotivation/Disempowerment

Motivation

- Motivation 1.0
  - Air, Water, Food, etc.
- Motivation 2.0
  - Give Rewards for desired behavior, punishments for undesired behavior
  - i.e. bonus of $50 to the employee who makes the most widgets this week.



https://www.flickr.com/photos/cdorobek/4408703896

Impacts to creative work

- Extrinsic Rewards (and punishments) negatively impact performance on creative tasks

Federal Reserve Study (MIT, CMU, UC)

- Variety of puzzles, unscrambling anagrams, recalling string of digits, etc.
- Group A: 1 Days Pay
- Group B: 2 Weeks Pay
- Group C: 5 Months Pay

- Group A/B about the same
- Group C worse in nearly every measure (8/9)

Worse results, long-term impacts

- Extrinsic rewards cause long-term impacts to performance of knowledge work.

Motivation 3.0

NORTHROP GRUMMAN

- Intrinsic Motivation
  - Autonomy
  - Mastery
  - Purpose
- Why would people give up significant (i.e. 20-30 hours a week) portion of time to update and add to an online encyclopedia?
- Why would developers (who have jobs) spend free time writing software?

Autonomy

- The ability to choose what you work on/do

Mastery

- The desire to get better at something

Purpose

- Doing something that transcends the self

Increase Intrinsic Motivation at Work

NORTHROP GRUMMAN

- Autonomy in daily work
- Scrum allows teams to commit to work for the sprint
- Team members choose what tasks they work on
- Team members choose how they do the work
- Team members choose their development tool

What Epics do the teams work on

- Similar types of work
  - Efficient
  - Predictable Velocity
  - Boring
  - Not Challenging

- Different types of work
  - Less efficient (esp. short term)
  - More experts on each kind of work
  - Challenging
  - Develop new skills
  - Redundancy of knowledge

Allow teams to choose their own epics

NORTHROP GRUMMAN

- All Teams send representatives (~2-3 including Scrum Master)
- Layout what the work to be accomplished is
  - Product Owner, Tech Lead, Customer, SME cover the work
  - Spell out restrictions.
- Rough Sizing of the work
- Teams distribute the work amongst themselves

THE VALUE OF PERFORMANCE.

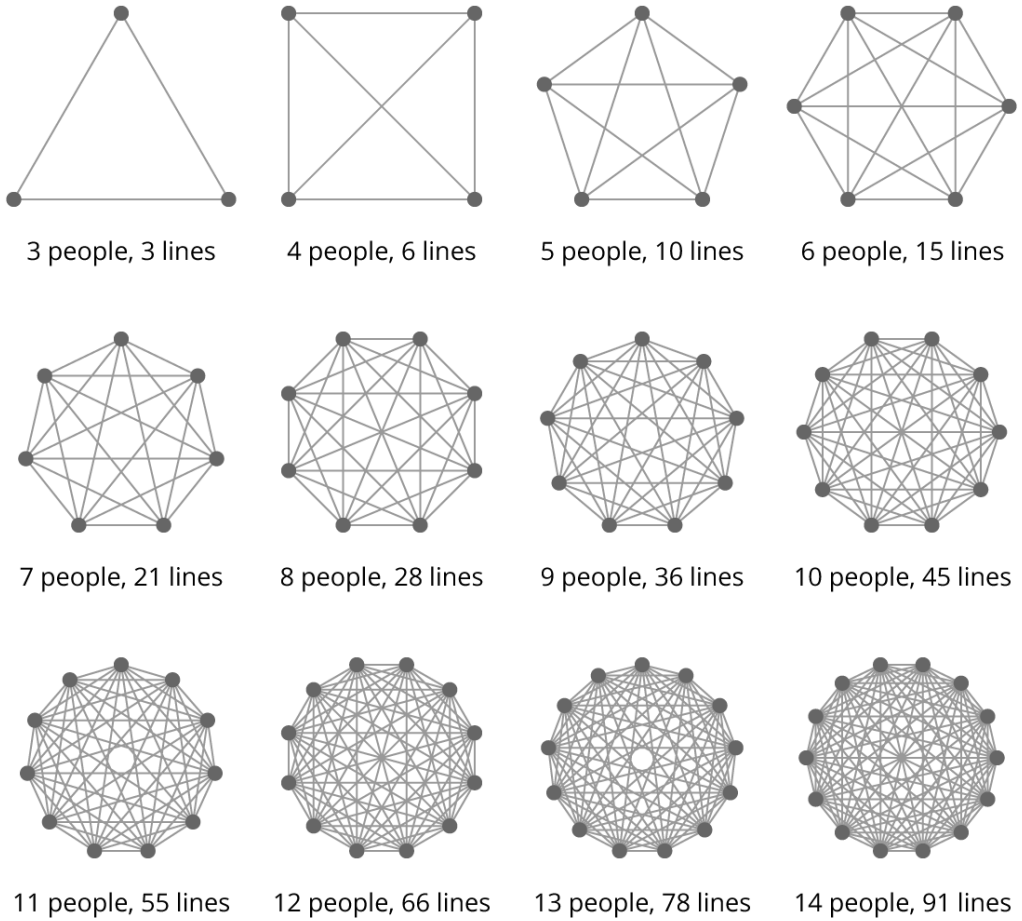
NORTHROP GRUMMAN

Broken Communication

NORTHROP GRUMMAN

Intra-Team Communication

Importance of Team Size

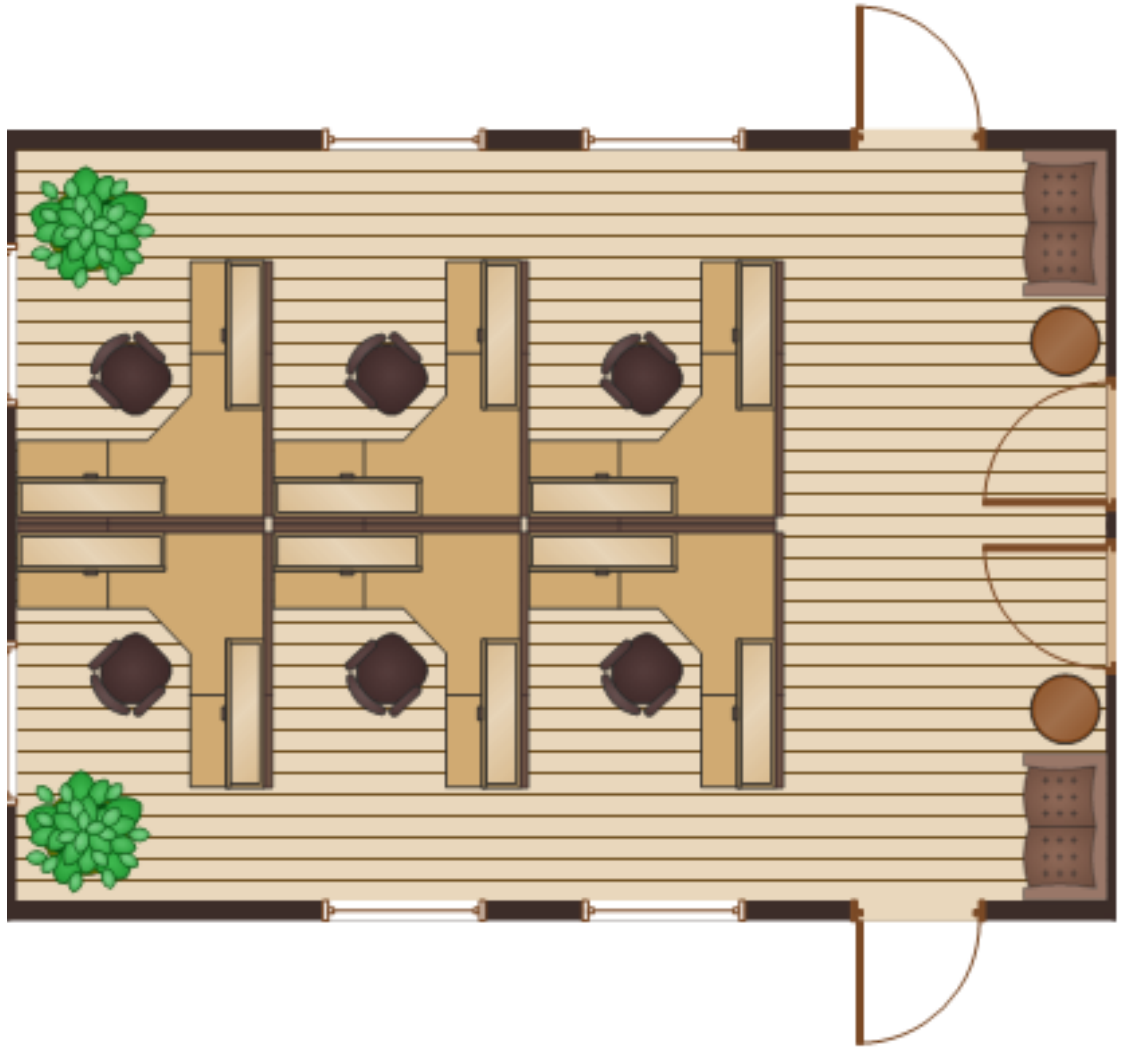


Office Layout

- What is the best setup in an office?

Colocation

- Teams are split by any barrier
- Effectively sub-teams



What is the best office design?

- Promote Autonomy by providing choices
- Variety of spaces
  - Open Layout
  - Heads Down Space
  - Collaboration rooms
  - Movable desks, partitions

- Open Office

Conway's Law

- "organizations which design systems ... are constrained to produce designs which are copies of the communication structures of these organizations"
- How can collaborative interactive software be created when development team members cannot communicate?

Inter-team Communication

- Colocated teams vs teams located all over building
- Exchange of Ideas between teams
  - CC/HB tight cohesion even years after colocation
  - GM/DB merged into team
  - Separate team remains more isolated

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Estimation

Hours Estimates

NORTHROP GRUMMAN

- Hours Estimates are mired in problems
- Eternally optimistic developers
- Jaded developers.
  - I Estimated 40 hours, took 80, got in trouble
  - I Estimated 80 hours, took 40, got in trouble
  - Solution: Estimate 80 hours take 80 (even if it only takes 40)

Abstract concepts i.e. Story points

NORTHROP GRUMMAN

- Why Story points?
- People are bad at estimating, but good at grouping
- Ease of preparing fruit for a fruit salad
  - Pineapple, grape, orange, apple, jackfruit

The secondary value of new (abstract) sizing

NORTHROP GRUMMAN

- We lost the fight on hours estimates
- We may yet be able to salvage Story points
  - Story point comparison between teams is invalid, don't do it.
  - Team Velocity should be judged based on variance, not on velocity
  - DO NOT USE SP's to punish/reward the team.

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

Responsibility for Problems

Agile Teams

- Team Members
- Facilitators (i.e. ScrumMasters/Coaches)
- Product Owners
- Managers

Team Members

Insecurity busting

NORTHROP GRUMMAN

- Creating a culture of continuous learning
- Remember: Autonomy, Mastery, Purpose create intrinsic motivation
- The number of Software Engineers is doubling every 5 years.
  - 10 years of experience means you have more experience than 75% of the Software Engineers
- Engineers are not Mushrooms, keeping them in the dark and feeding them crap all day is not a reasonable approach

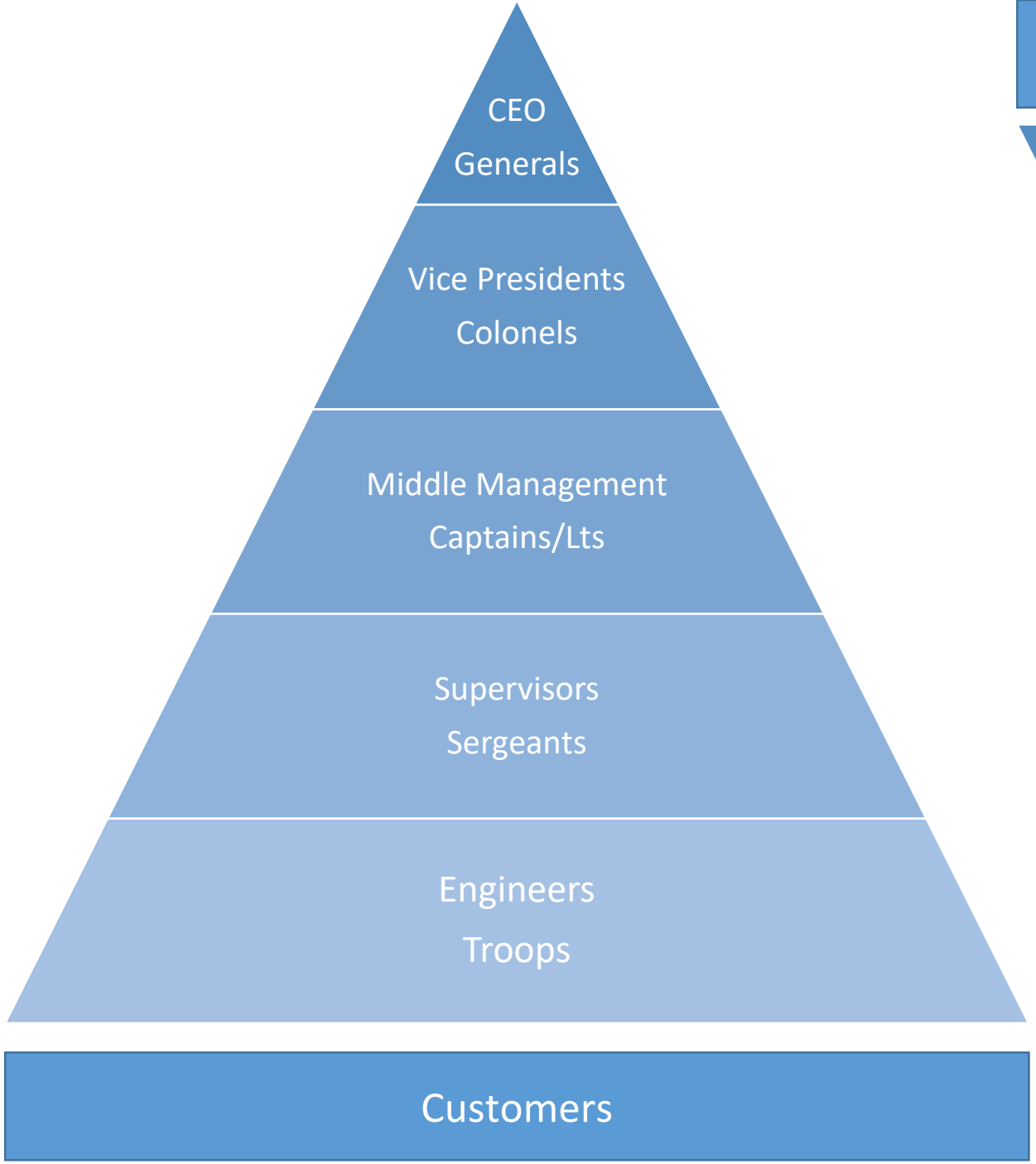
Management

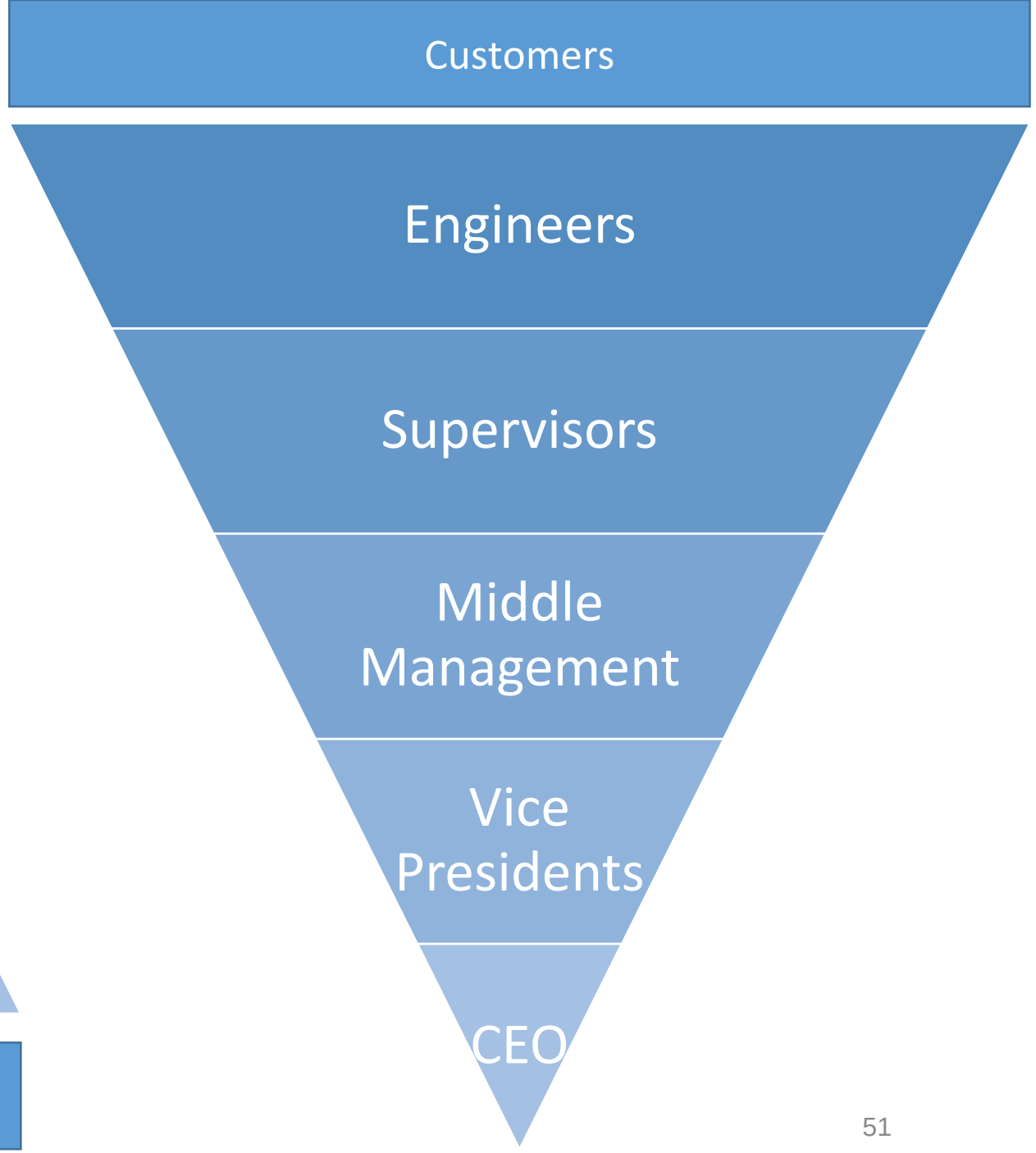
Leadership

- Who Leads a Software team?
  - ScrumMaster
  - Product Owner
  - Manager
  - Team Members
- Leaders are at all levels,
  - If managers start calling themselves “leaders” then only they can lead.

Leadership at all levels

- Mentor/Mentee relationships (formal or informal)
- Pair programming
- Hands off keyboard for a sprint
- Allow knowledge to flourish
- Take leadership where it is found, don’t discriminate based on experience





Top Down Command Structure

NORTHROP GRUMMAN

- The Enemy is at the bottom
- “need people from the neck down”
- Works for repetitive non-thinking work.
- Software Engineers are needed from the neck up, creative (remember intrinsic motivation)

Inverted Org Chart/Servant Leadership

- Engineer
- SM/PO
- Manager

- Send the message, We produce/sell software.
  - The most important people in the organization are the engineers producing a marketable product, all others in the organization need to support that goal.
- “The Servant” - Hunter

Facilitators

Team messaging problems

NORTHROP GRUMMAN

- Cross-functional != Anyone on team can/will do everything.
  - I'm a Sys Admin, there's no way I can/will write a Selenium test for a new component.
- Story points are not meaningless, they mean whatever the team decides they mean.
- Velocity is not a metric that we care about

Product Owners

Rapid Shifts

NORTHROP GRUMMAN

- PO prioritizes work
- PO coordinates between stakeholders and team
- Require enough capacity to not be causing churn within the team
  - Scrum is “Just in time” and good at “responding to change”, however it’s best to not surprise a team mid sprint with a change in direction
- 50/50 time spent with team vs time spent with stakeholders.

Summary

- Honesty
- Autonomy, Mastery, Purpose (Motivation 3.0)
- Remove Impediments to communication
- Build Trust
- Build Leaders
- Act with Intent, don’t just react to problems

Contact Information

- https://www.meetup.com/Agile-for-Defense/
- https://www.linkedin.com/in/scott-grimes-792277a/
- Scott.Grimes@ngc.com

THE VALUE OF PERFORMANCE.

NORTHROP GRUMMAN

- It is a simple thing to disrupt a software team, bad management can fracture a team, however significant damage can be done by team members themselves. Recovery from inter-team dysfunctions can feel impossible to solve. It is important to know these pitfalls and how to avoid them. Teams at all levels of maturity can be disrupted, the dysfunctions of disempowerment and broken communication can quickly damage the productivity of any team, but these disruptions will be especially felt in an agile software team reliant on interactions for success. What factors have the largest impact on teams? And what can team members, facilitators, and managers do to strengthen teams and avoid the pitfalls? This presentation will cover real world team problems encountered in software development projects and what specific actions were taken to fix these problems from within teams.