

Week: COMP 2120 / COMP 6120
6 of 12

TEAM CULTURE

A/Prof Alex Potanin



ANU Acknowledgment of Country



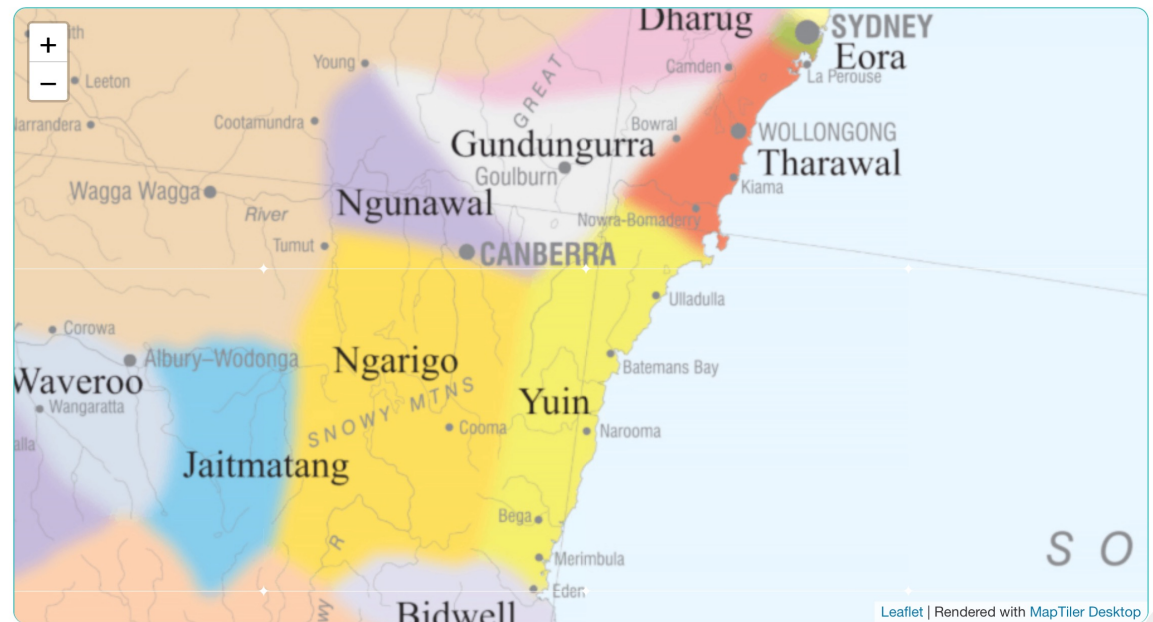
AIATSIS

Explore Family history Collection Research Education What's new About

Shop

Search

“We acknowledge and celebrate the First Australians on whose traditional lands we meet, and pay our respect to the elders past and present.”



<https://aiatsis.gov.au/explore/map-indigenous-australia>



Today

- 10X Engineers
- Teams and Team Issues
- Modern Team Structures
- Virtual Teams
- General Guidelines
- Developer Turnover and Motivation
- Managing Programmers
- Documentation



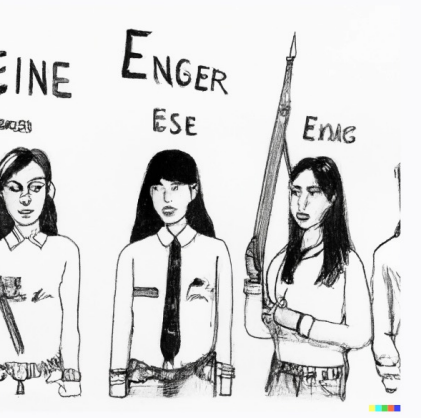
Edit the detailed description

Surprise me

Upload →

pencil drawing of 10X Engineers and rock stars with no words

Generate



10X Engineers

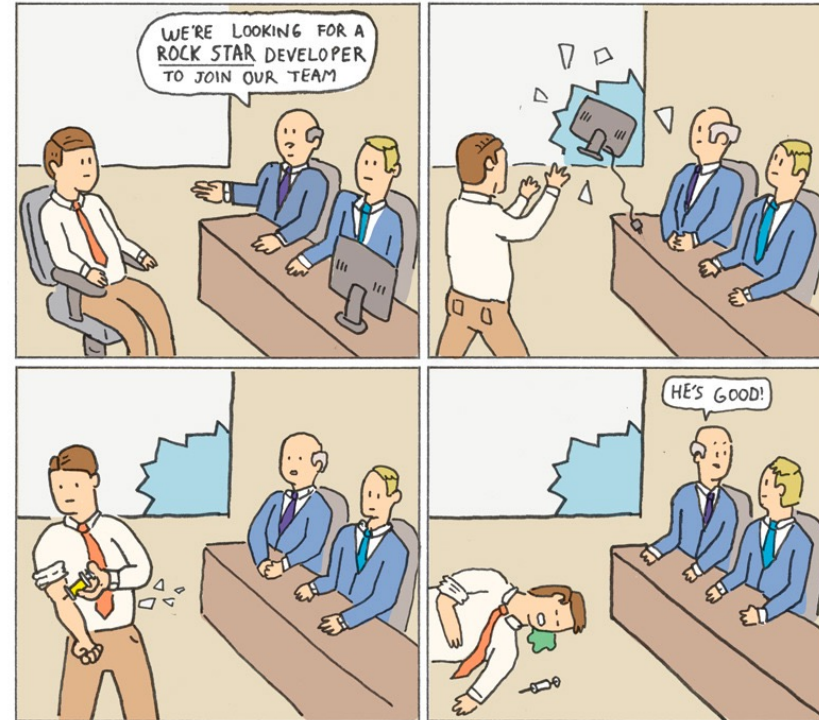


10X Engineers

- Aka “rock-star”, “ninja”



ROCK STAR DEVELOPER

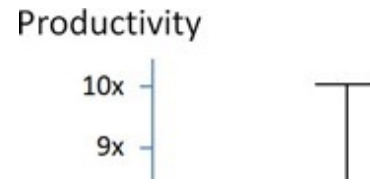


@SKELETON_CLAW

SKELETONCLAW.COM

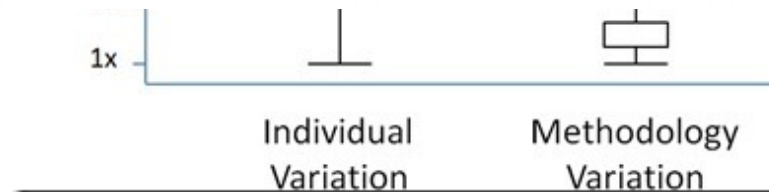


1966 study on online/offline programming performance



Performance variables. To paraphrase a nursery rhyme:

When a programmer is good,
He is very, very good,
But when he is bad,
He is horrid.



<https://www.construx.com/blog/the-origins-of-10x-how-valid-is-the-underlying-research/>



10x




- Reported as early as 1968 (Sackman, Erickson, and Grant)
 - Coding time 20:1
 - Debugging time 25:1
 - Program size 5:1
 - Execution speed 10:1
 - No correlation to amount of experience
- "order-of-magnitude differences among programmers" repeatedly reported
- Differences not explained by
 - programming language
 - years of experience



Poll Everywhere Time!

Join by Web PollEv.com/potantin Join by Text Send [potantin](https://poll-ev.com/potantin) to 22333



Are YOU a 10X Engineer? 👍 0

Yes (A)

No (B)





ER #00120C

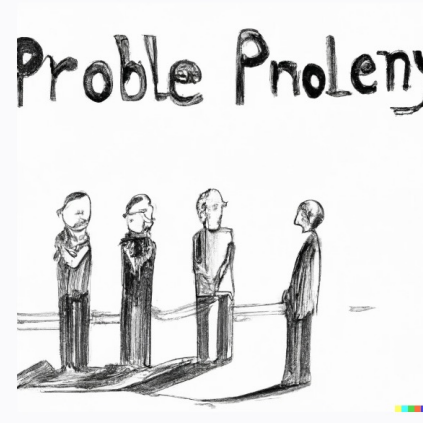
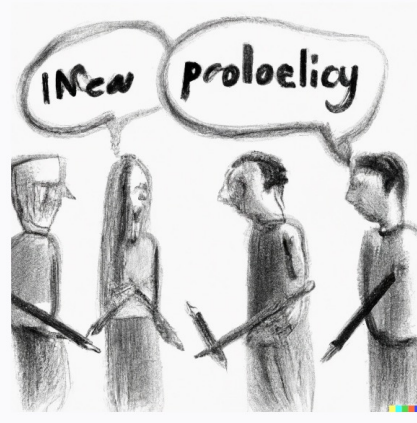


Edit the detailed description

Surprise me Upload →

pencil drawing of a team having problems with no words

Generate



Teams and Team Issues

Necessity of Groups

- Division of labor
- Division of expertise (e.g., security expert, database expert)



Team Issues

1. Social Loafing
2. Groupthink
3. Multiple/Conflicting Goals
4. Process Costs

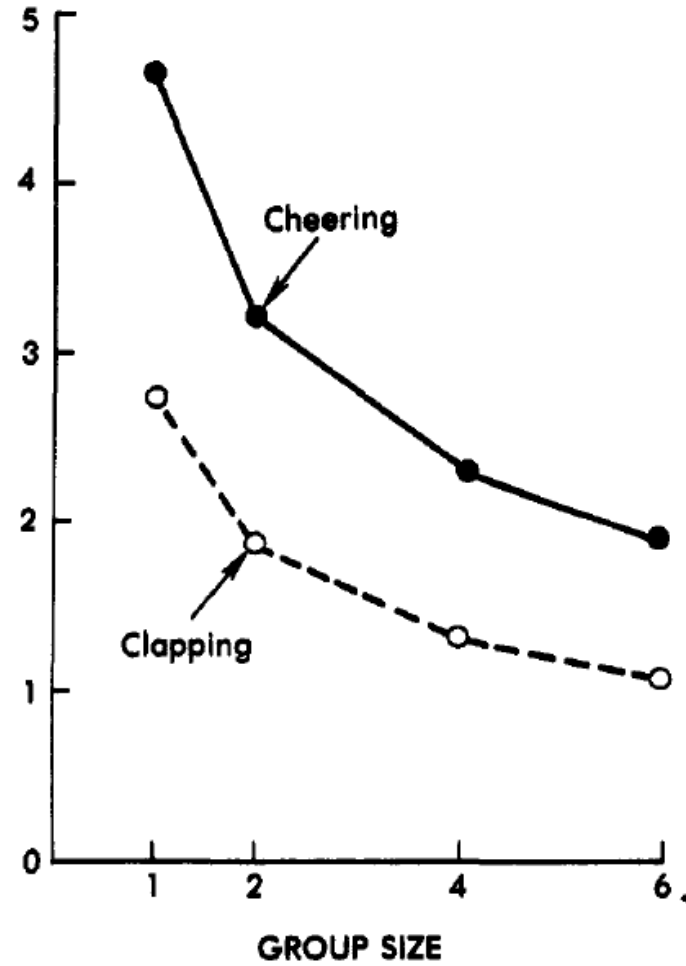


Team Issue 1: Social Loafing





SOUND PRESSURE PER PERSON IN DYNES PER cm^2



Latane, Bibb, Kipling Williams, and Stephen Harkins. "Many hands make light the work: The causes and consequences of social loafing." *Journal of personality and social psychology* 37.6 (1979): 822.



Social loafing



- People exerting less effort within a group
- Reasons
 - Diffusion of responsibility
 - Motivation
 - Dispensability of effort / missing recognition
 - Avoid pulling everybody / "sucker effect"
 - Submaximal goal setting
- “Evaluation potential, expectations of co-worker performance, task meaningfulness, and culture had especially strong influence”

Karau, Steven J., and Kipling D. Williams. "Social loafing: A meta-analytic review and theoretical integration." *Journal of personality and social psychology* 65.4 (1993): 681.



Mitigation Strategies



- Involve all team members, co-location
- Assign specific tasks with individual responsibility
 - Increase identifiability
 - Team contracts, measurement
- Provide choices in selecting tasks
- Promote involvement, challenge developers
- Reviews and feedback
- Team cohesion, team forming exercises
- Small teams



Responsibilities & Buy-In

- Involve team members in decision making
- Assign responsibilities (ideally goals not tasks)
- Record decisions and commitments; make record available



Team Issue 2: Groupthink



Groupthink

- Group minimizing conflict
- Avoid exploring alternatives
- Suppressing dissenting views
- Isolating from outside influences
- -> Irrational/dysfunctional decision making





Star Wars: Episode I - The Phantom Menace (1999)

🌟 55% 🍿 59%

Critics Consensus: Burdened by exposition and populated with stock characters, *The Phantom Menace* gets the *Star Wars* prequels off to a bumpy – albeit visually dazzling – start.

Starring: Liam Neeson, Ewan McGregor, Natalie Portman

Director: George Lucas



Star Wars: Episode VI - Return of the Jedi (1983)

🌟 80% 🍿 94%

Critics Consensus: Though failing to reach the cinematic heights of its predecessors, *Return of the Jedi* remains an entertaining sci-fi adventure and a fitting end to the classic trilogy.

Starring: Mark Hamill, Carrie Fisher, Harrison Ford

Director: Richard Marquand



Star Wars: Episode V - The Empire Strikes Back (1980)

🌟 95% 🍿 97%

Critics Consensus: Dark, sinister, but ultimately even more involving than *A New Hope*, *The Empire Strikes Back* defies viewer expectations and takes the series to heightened emotional levels.

Starring: Mark Hamill, Harrison Ford, Carrie Fisher

Director: Irvin Kershner



Star Wars: Episode IV - A New Hope (1977)

🌟 93% 🍿 96%

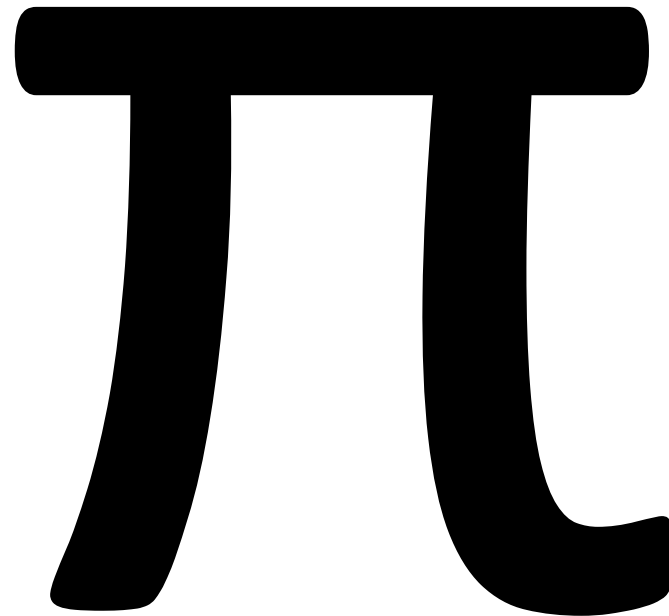
Critics Consensus: A legendarily expansive and ambitious start to the sci-fi saga, George Lucas opened our eyes to the possibilities of blockbuster filmmaking and things have never been the same.

Starring: Mark Hamill, Harrison Ford, Carrie Fisher

Director: George Lucas



Time and Cost Estimation



Causes of Groupthink

- High group cohesiveness, homogeneity
- Structural faults (insulation, biased leadership, lack of methodological exploration)
- Situational context (stressful external threats, recent failures, moral dilemmas)



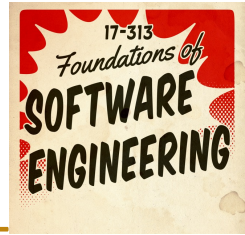
Symptoms

- **Overestimation of ability**
 - invulnerability, unquestioned believe in morality
- **Closed-mindedness**
 - ignore warnings, stereotyping
 - innovation averse
- **Pressure toward uniformity**
 - self-censorship, illusion of unanimity, ...





Diversity



“Men and women have different viewpoints, ideas, and market insights, which enables better **problem solving**. A gender-diverse workforce provides easier **access to resources**, such as various sources of credit, multiple sources of information, and wider industry knowledge. A gender-diverse workforce allows the company to **serve an increasingly diverse customer base**. Gender diversity helps companies **attract and retain talented women**.”

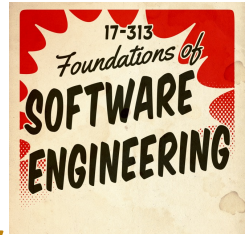
“Cultural diversity leads to **process losses** through task conflict and decreased social integration, but to **process gains** through increased creativity and satisfaction.”

<http://www.gallup.com/businessjournal/166220/business-benefits-gender-diversity.aspx>

Stahl, Günter K., et al. "Unraveling the effects of cultural diversity in teams: A meta-analysis of research on multicultural work groups." *Journal of international business studies* 41.4 (2010): 690-709.



Studies Show



- Gender-diverse management teams showed superior return on equity, debt/equity ratios, price/equity ratios, and average growth. -Rohner, U. and B. Dougan (2012)
- Gender-balanced teams were the most likely to experiment, be creative, share knowledge, and fulfill tasks. -Lehman Brothers Center for Women in Business. (2008)
- Gender diversity on technical work teams was associated with superior adherence to project schedules, lower project costs, higher employee performance ratings, and higher employee pay bonuses. -Turner, L. (2009)



Unconscious Bias



We all have shortcuts, or “schemas,” that help us make sense of the world. But our shortcuts sometimes make us misinterpret or miss things. That’s **unconscious bias**.



Unconscious Bias

- Pervasive, cultural
- Raise awareness
- Explicit goals
- Measurement



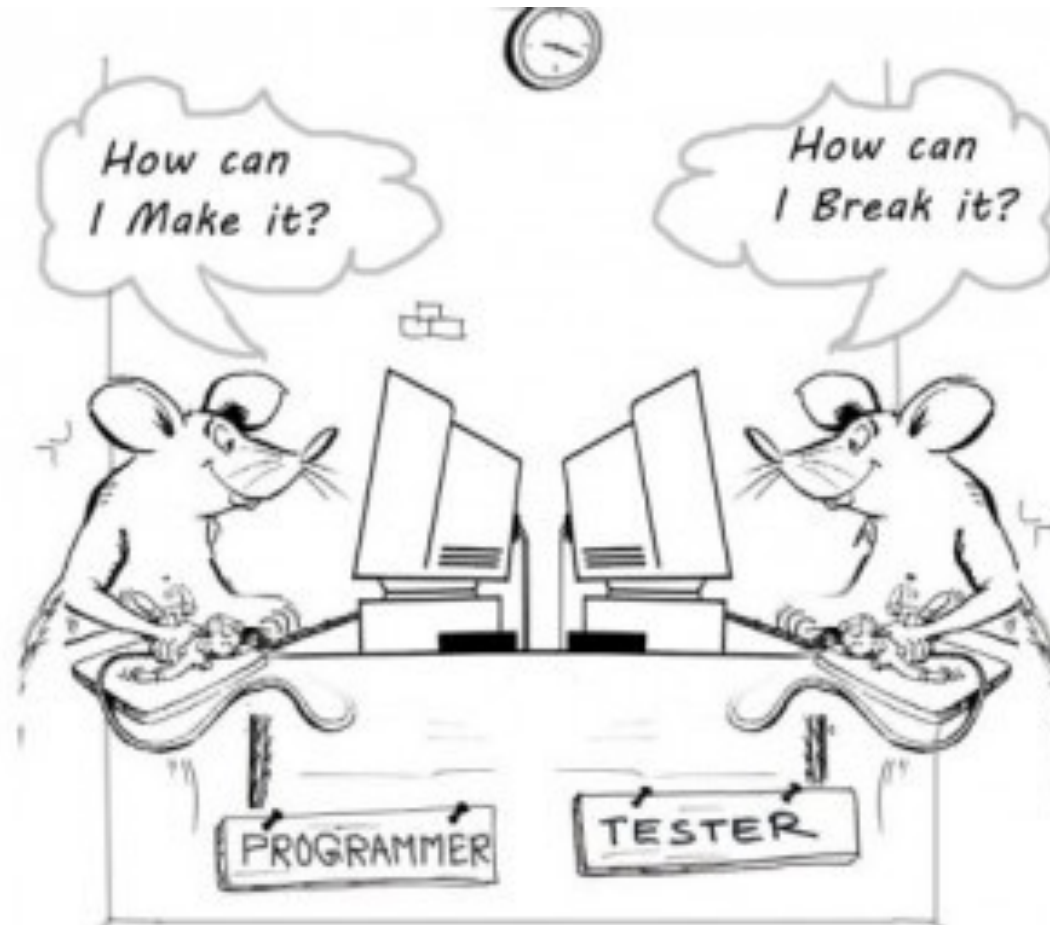
Mitigation Strategies



- Several agile techniques
 - Planning poker (<https://www.mountangoatsoftware.com/agile/planning-poker>)
 - Tests, continuous integration
 - On-site customers
- Diverse teams
- Management style
- Avoid HR evaluation by metrics
- Separate QA from development
- Outside experts
- Process reflection
- ...

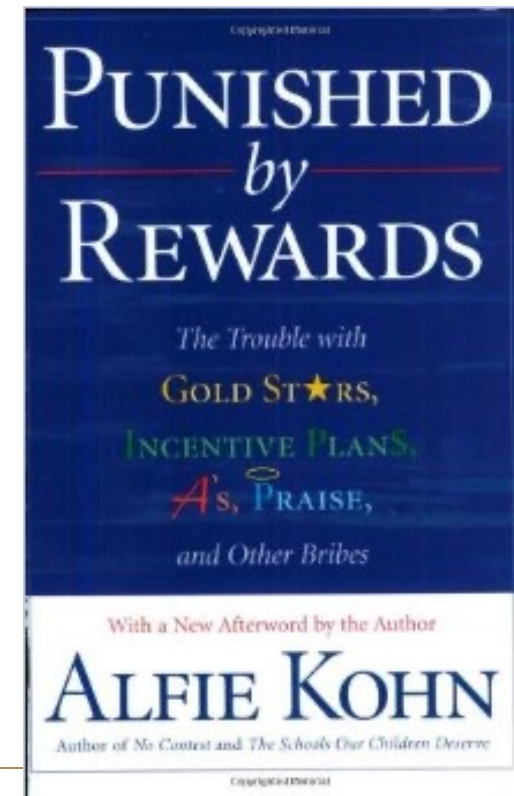


Team Issue 3: Multiple/Conflicting Goals

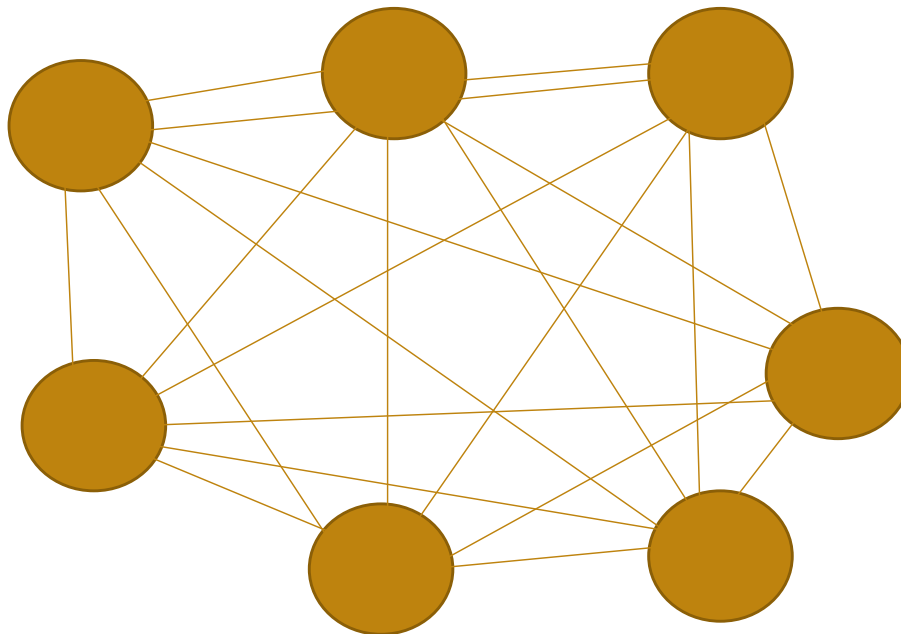


Incentives?

- Team incentives
- vs individual incentives?



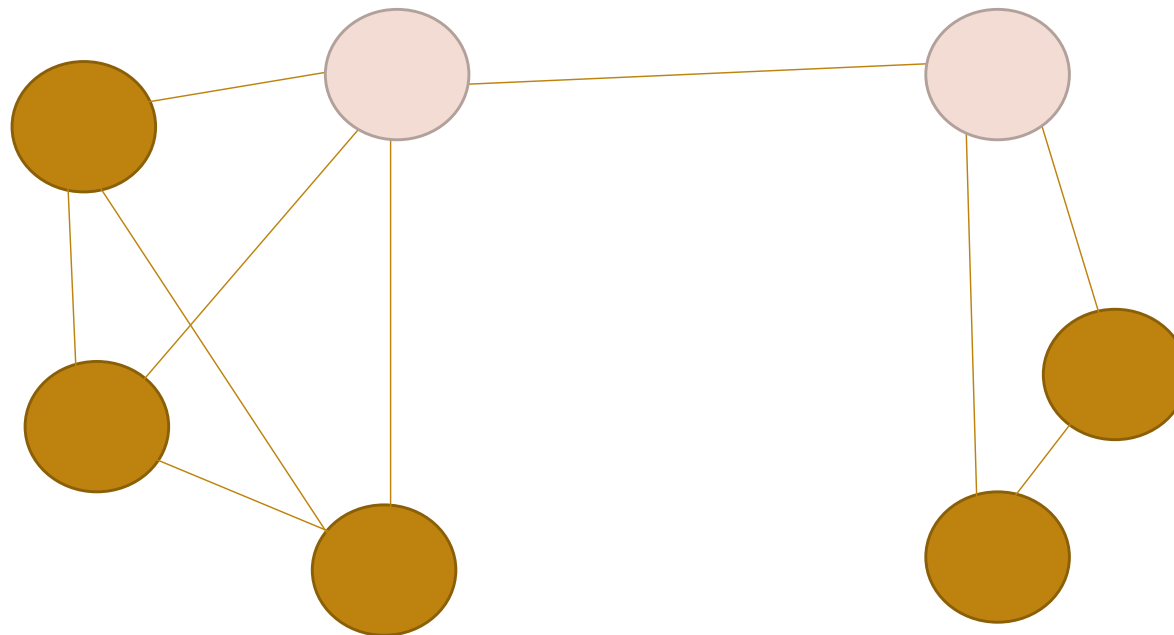
Team Issue 4: Process Costs



$n(n - 1) / 2$
communication links



Team Issue 4: Process Costs



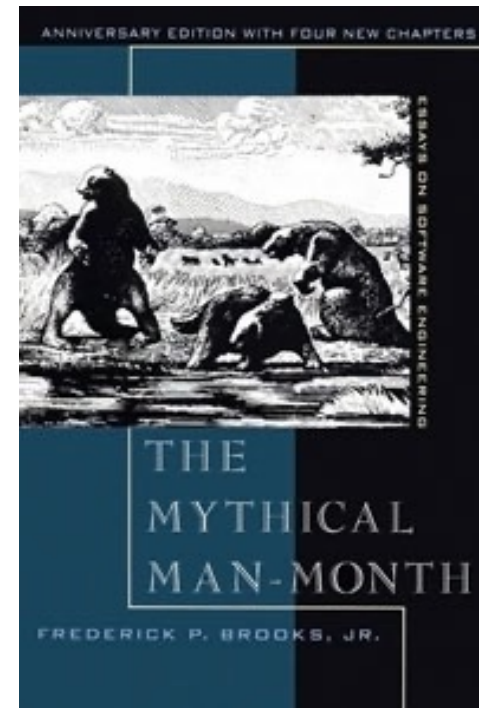
Mythical Man Month




- Brooks's law: *Adding manpower to a late software project makes it later*


ACTIVITY: Discuss reasons in groups.

1975, describing experience at IBM developing OS/360



Poll Everywhere Time!

Join by Web PollEv.com/potantin Join by Text Send [potantin](tel:22333) to 22333 

Which of these was most surprising to you today?  0

Social Loafing **(A)**

Groupthink **(B)**

Multiple/Conflicting Goals **(C)**

Process Costs **(D)**





ER #00120C

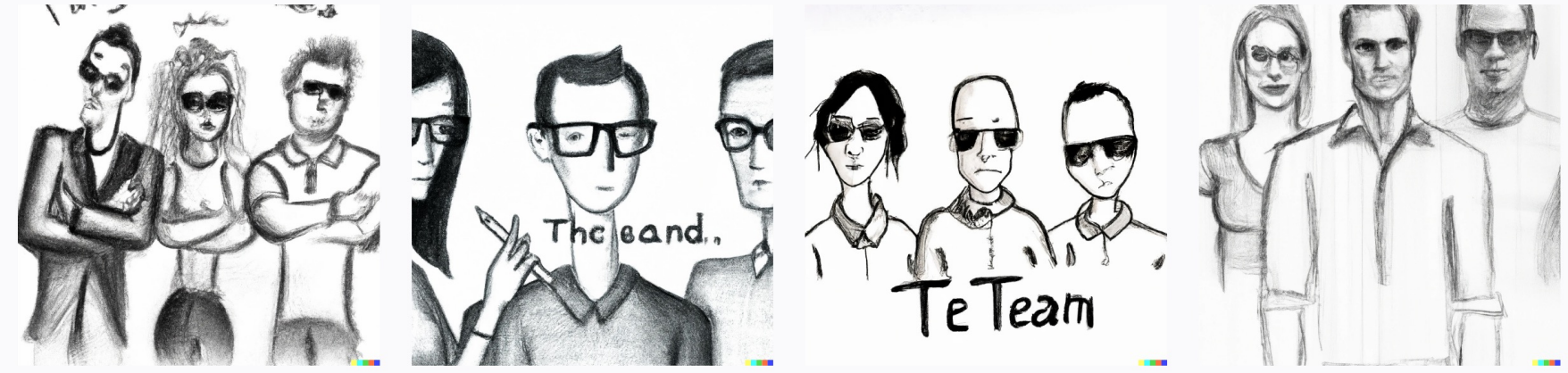


Edit the detailed description

Surprise me Upload →

pencil drawing of a modern cool team with no words

Generate



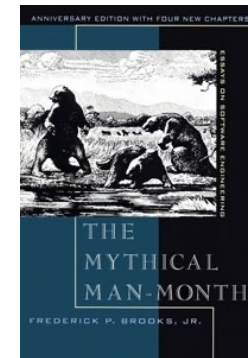
Modern Team Structures

Brook's Surgical Teams



- Chief programmer – most programming and initial documentation
- Support staff
 - Copilot: supports chief programmer in development tasks, represents team at meetings
 - Administrator: manages people, hardware and other resources
 - Editor: editing documentation
 - Two secretaries: one each for the administrator and editor
 - Program clerk: keeps records of source code and documentation
 - Toolsmith: builds specialized programming tools
 - Tester: develops and runs tests
 - Language lawyer: expert in programming languages, provides advice on producing optimal code.

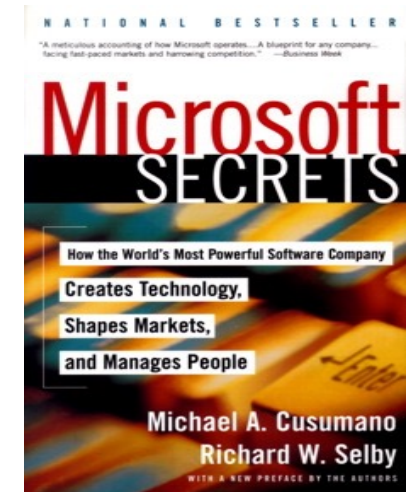
IBM 1971



Microsoft's Small Team Practices



- Vision statement and milestones (2-4 month), no formal spec
- Feature selection, prioritized by market, assigned to milestones
- Modular architecture
 - Allows small federated teams (Conway's Law – slide coming up)
- Small teams of overlapping functional specialists



Windows 95: 200 developers and testers, one of 250 products

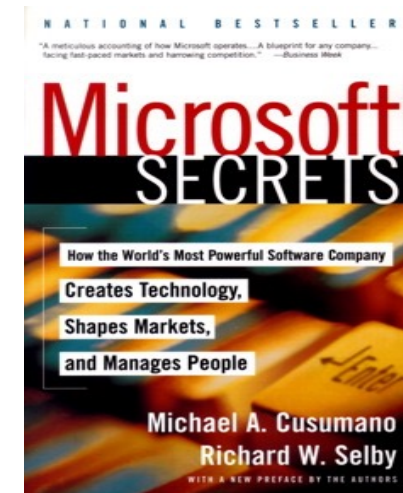


Microsoft's Small Team Practices



- **Feature Team**

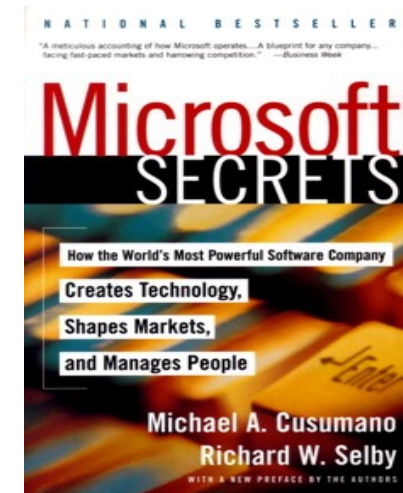
- 3-8 developers (design, develop)
- 3-8 testers (validation, verification, usability, market analysis)
- 1 program manager (vision, schedule communication; leader, facilitator) – working on several features
- 1 product manager (marketing research, plan, betas)



Microsoft's Small Team Practices



- "Synchronize and stabilize"
- For each milestone
 - 6-10 weeks feature development and continuous testing
 - frequent merges, daily builds
 - 2-5 weeks integration and testing ("zero-bug release", external betas)
 - 2-5 weeks buffer



Amazon Teams



Agile Practices (e.g., Scrum)

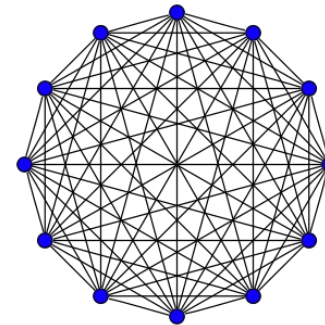
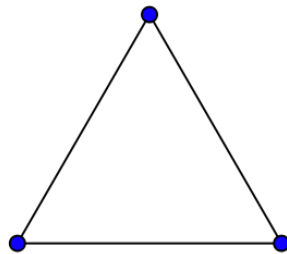
- 7+/-2 team members, collocated
- Self managing
- Scrum master (rotating role)
- Product owner / customer representative





Large teams (29 people) create around six times as many defects as small teams (3 people) and obviously burn through a lot more money. Yet, the large team appears to produce about the same amount of output in only an average of 12 days' less time. This is a truly astonishing finding, through it fits with my personal experience on projects over 35 years.

- Phillip Amour, 2006, CACM 49:9



Establish communication patterns

- Avoid overhead
- Ensure reliability
- Constraint latency

- e.g. Issue tracker vs email; online vs face to face



Design
opportunity



Awareness

- Notifications
- Brook's documentation book
- Email to all
- Code reviews



Conway's Law

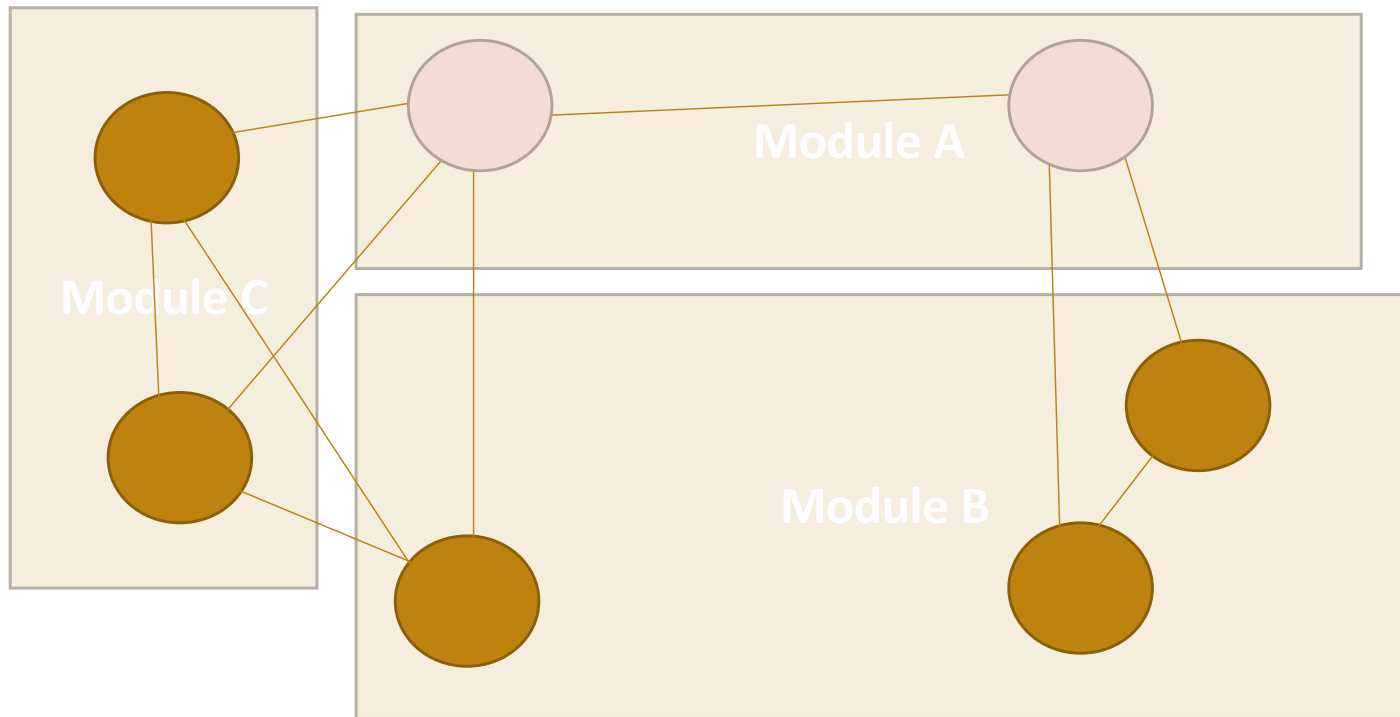
“Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization's communication structure.”

— *Mel Conway, 1967*

“If you have four groups working on a compiler, you'll get a 4-pass compiler.”



Congruence



Socio-Technical Congruence

- Structural congruence
- Geographical congruence
- Task congruence
- IRC communication congruence

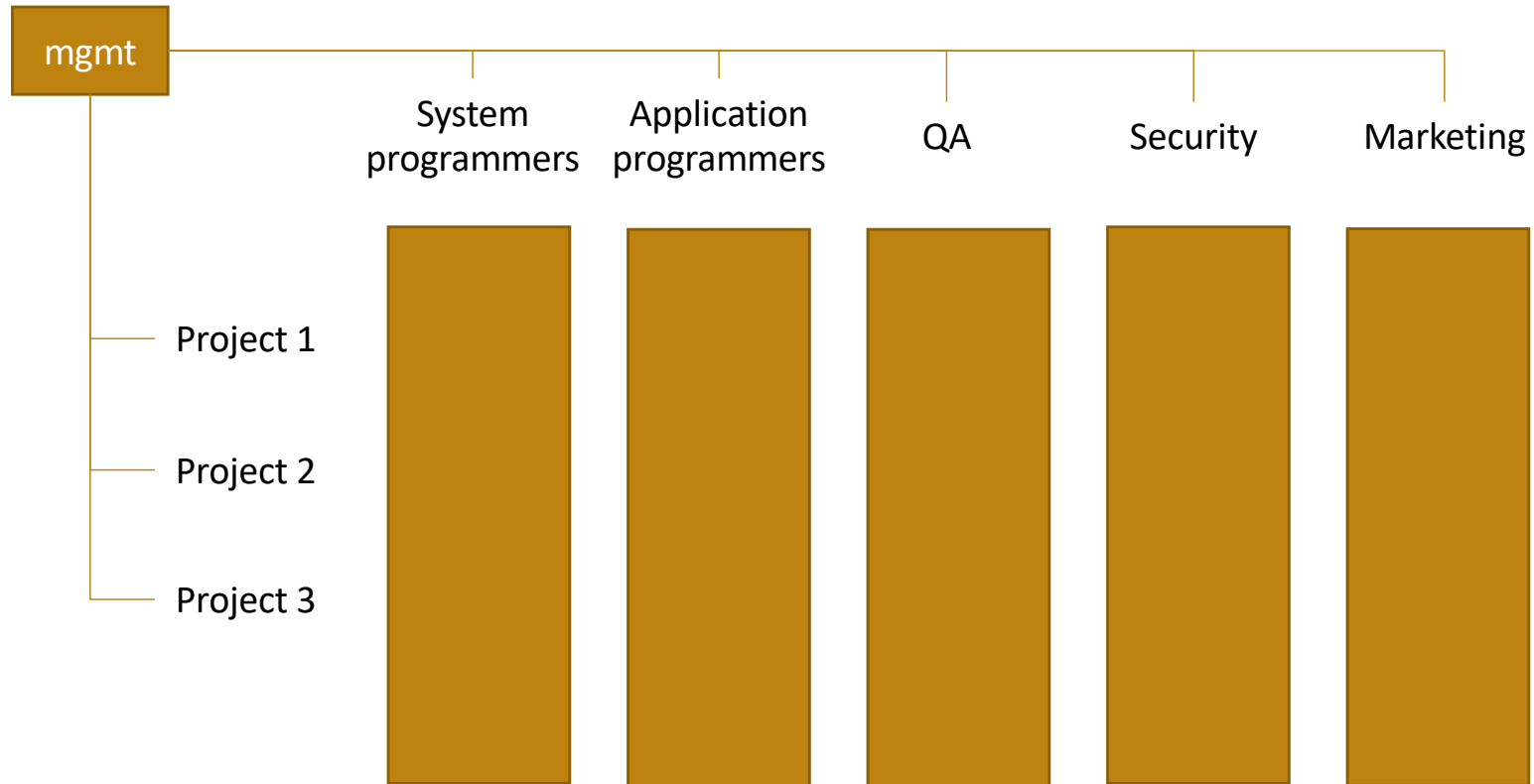


Teamwork Guidelines

- Respect Conway's Law
 - Code structure and team structure should align
- Seek well-defined, stable interfaces



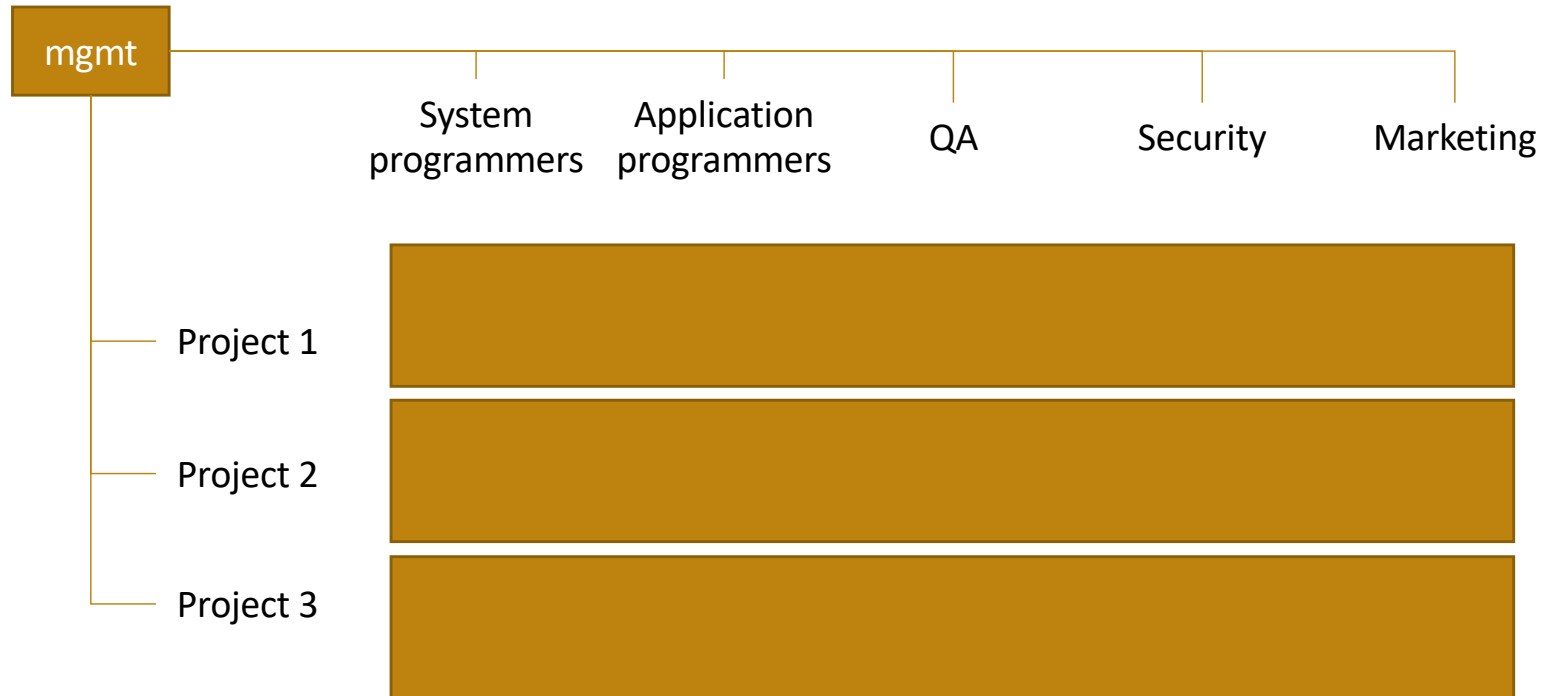
Matrix Organization



Temporary assignment to projects; flexible staffing



Project Organization



Case Study



Case Study: Brøderbund



- As the functional departments grew, staffing the heavily matrixed projects became more and more of a nightmare. To address this, the company reorganized itself into “Studios”, each with dedicated resources for each of the major functional areas reporting up to a Studio manager. Given direct responsibility for performance and compensation, Studio managers could allocate resources freely.
- The Studios were able to exert more direct control on the projects and team members, but not without a cost. The major problem that emerged from Brøderbund’s Studio reorganization was that members of the various functional disciplines began to lose touch with their functional counterparts. Experience wasn’t shared as easily. Over time, duplicate effort began to appear.



Commitment & Accountability



- Conflict is useful, expose all views
- Come to decision, commit to it
- Assign responsibilities
- Record decisions and commitments; make record available



Bell & Hart – 8 Causes of Conflict





- Conflicting resources.
- Conflicting styles.
- Conflicting perceptions.
- Conflicting goals.
- Conflicting pressures.
- Conflicting roles.
- Different personal values.
- Unpredictable policies.

Bell, Art. (2002). Six ways to resolve workplace conflicts.
McLaren School of Business, University of San Francisco.
<https://www.mindtools.com/pages/article/eight-causes-conflict.htm>



Poll Everywhere Time!

Join by Web PollEv.com/potantin Join by Text Send **potantin** to **22333** 


What is your favourite team structure from those you saw today? 

Join by Web
PollEv.com/potantin

Join by Text
Send **potantin** to **22333**

Join by QR code

Scan with your camera app





IS PROVIDER #00120C

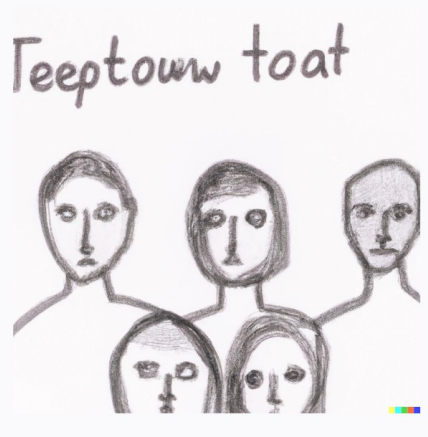


Edit the detailed description

Surprise me Upload →

pencil drawing of a virtual team with no words

Generate



Virtual Teams

Computer Supported Collaborative Work (CSCW): Technology-assisted collaboration



- Many failures
- Isolated, but very significant, success
 - Jazz, Github, ...





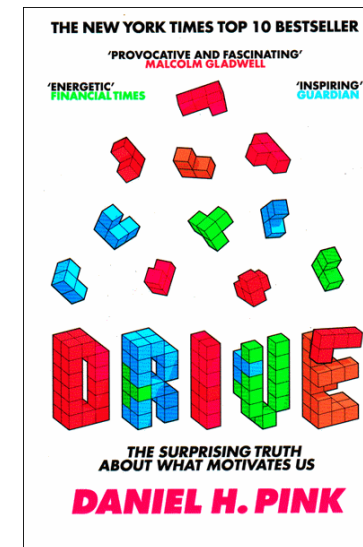
Spotify Squads

<https://www.atlassian.com/agile/agile-at-scale/spotify>

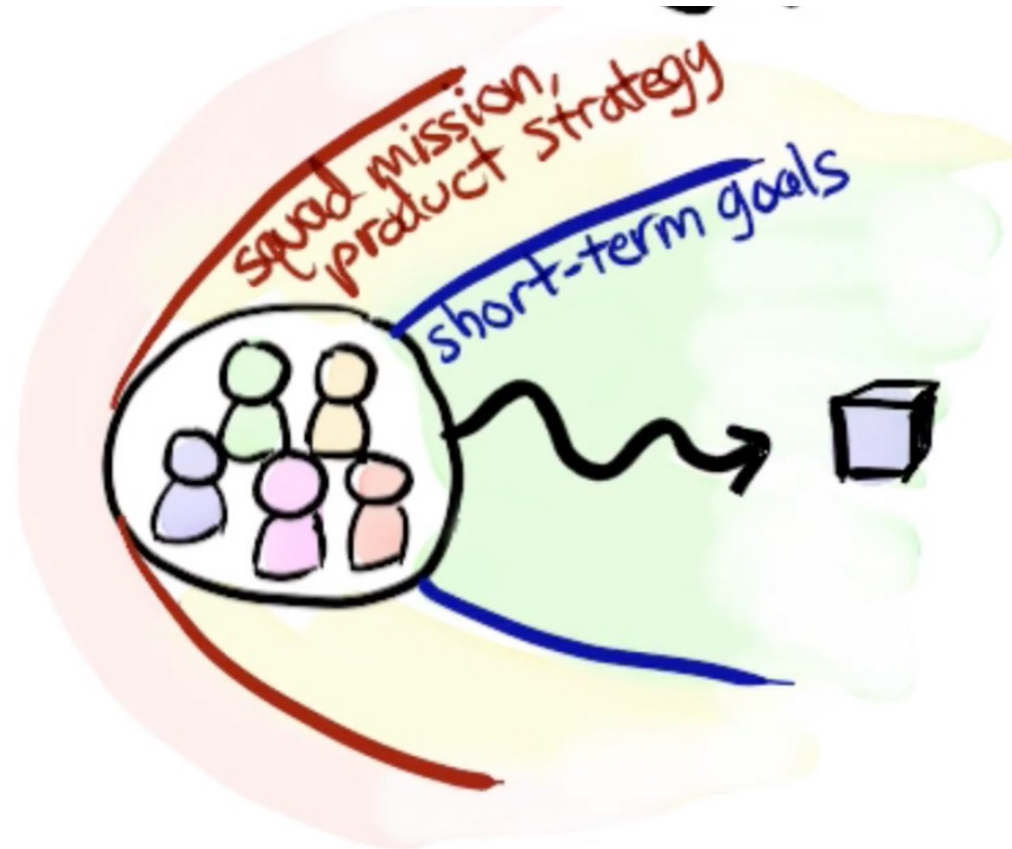


Principles

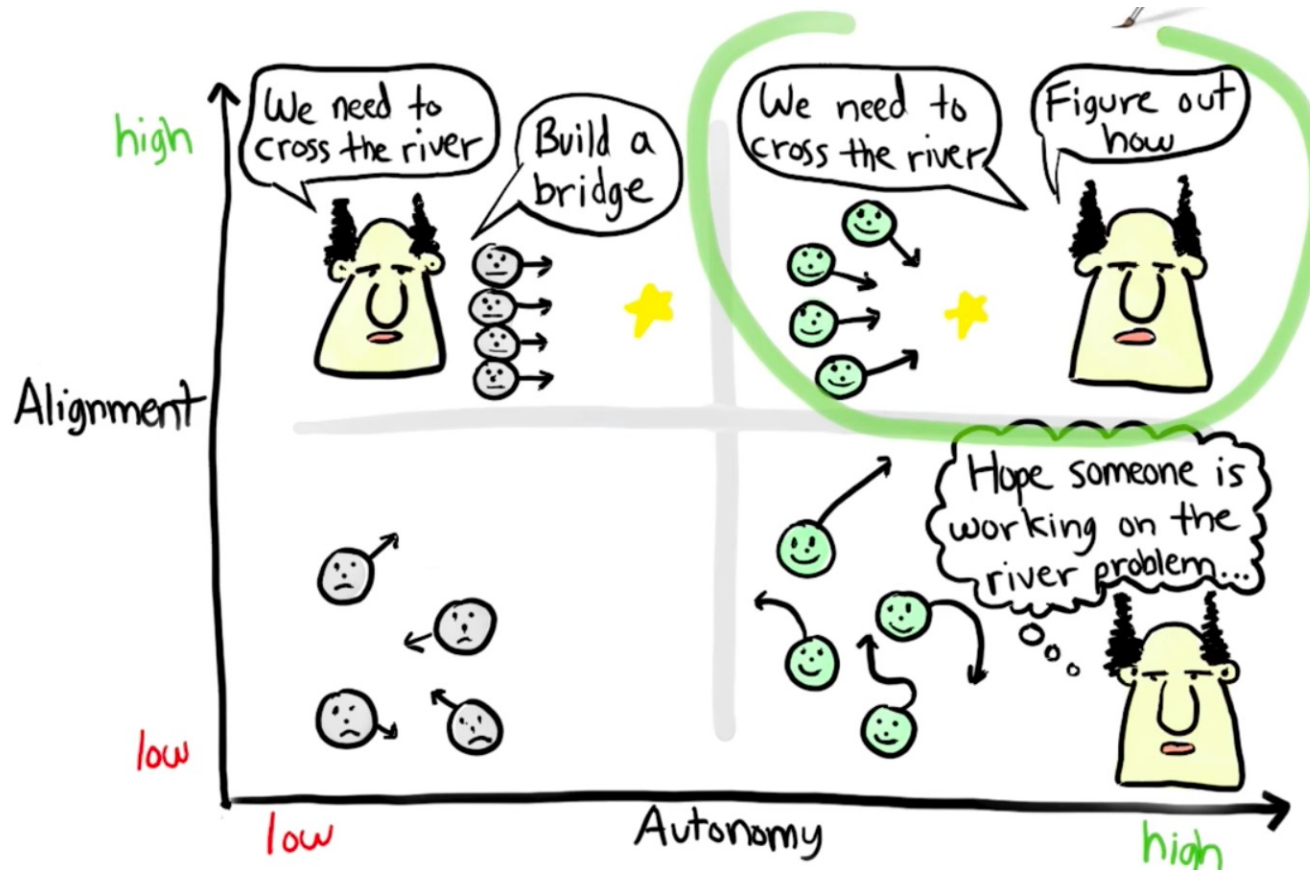
- Rules are a good start, then break them when needed
- Agile > Scrum
- Principles > Practices
- Autonomy, Mastery, Purpose
- *Be autonomous, but don't sub-optimize!*



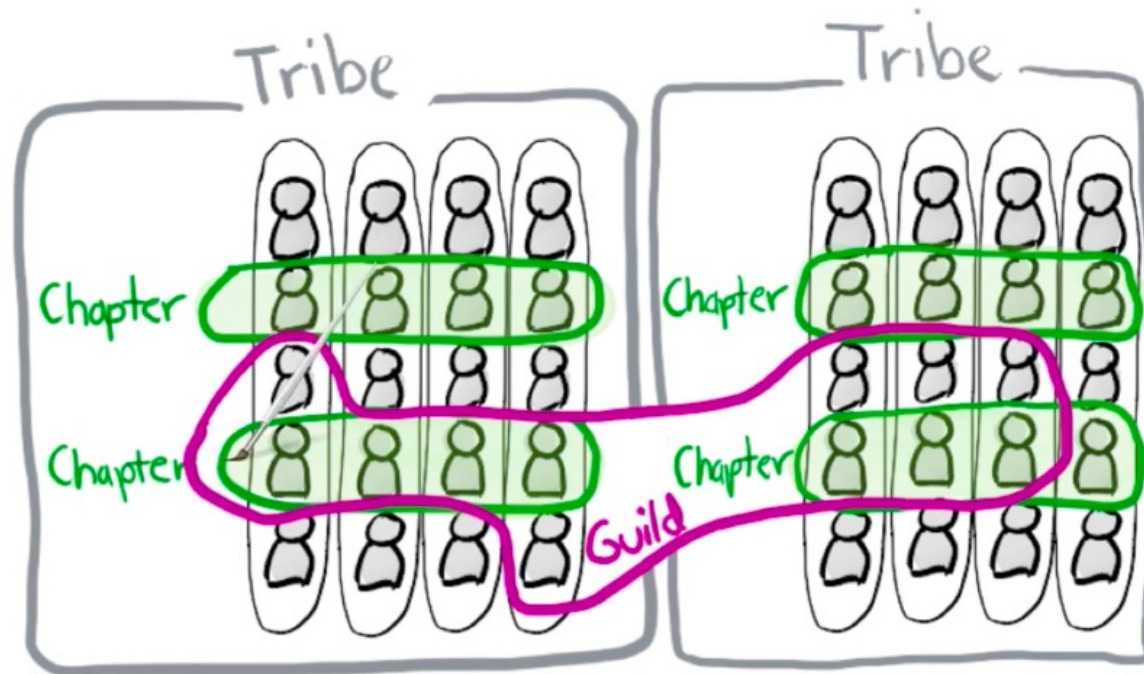
Autonomous Squads



Aligned Autonomous squads



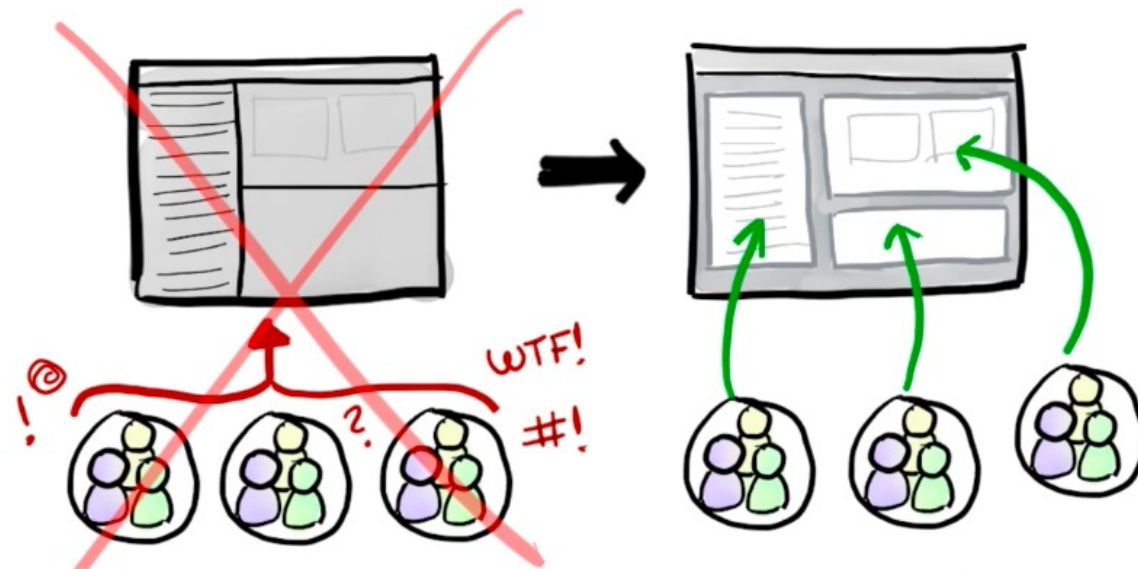
Squads, Tribes, Chapters, Guilds



Getting into production



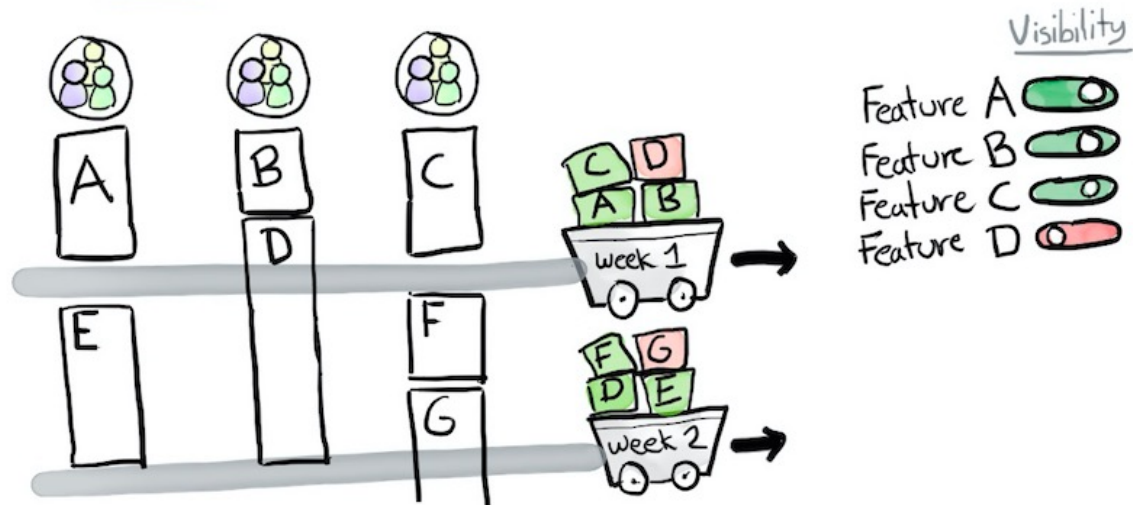
Decoupled releases



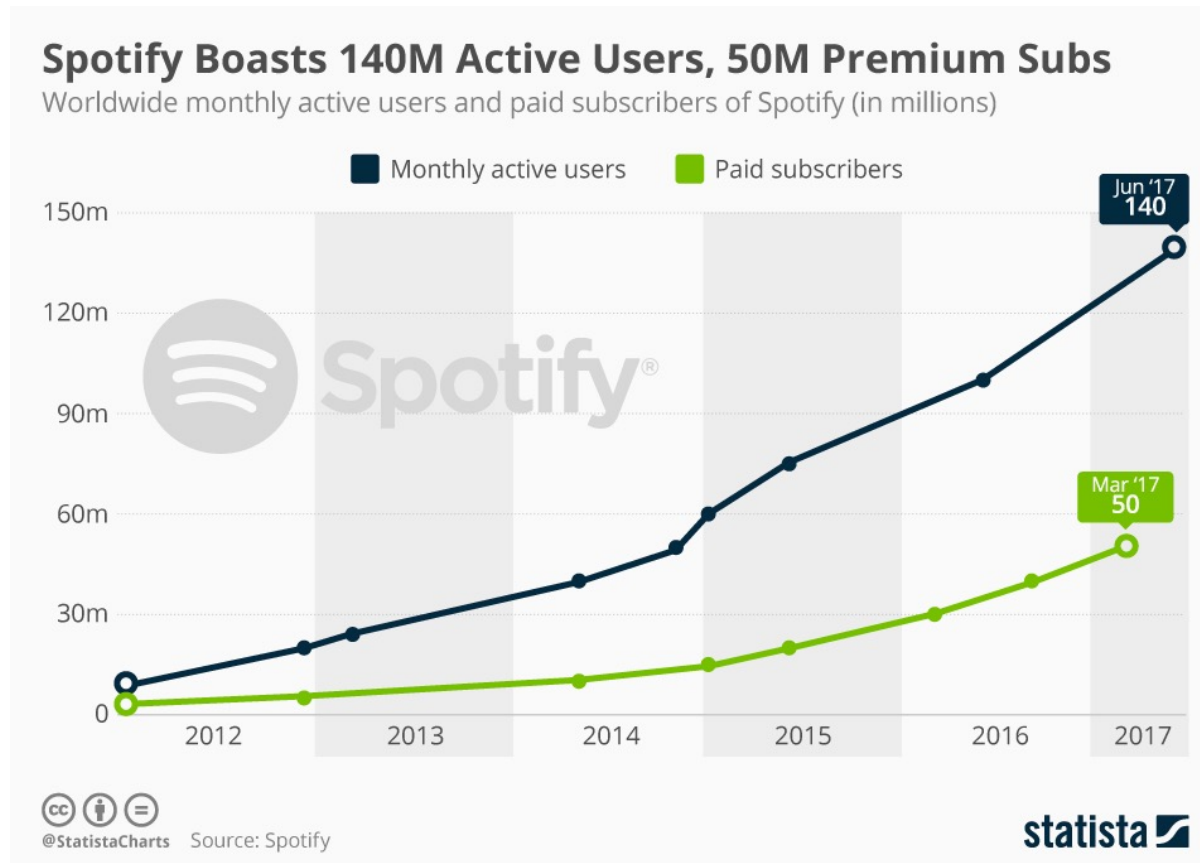
Decouple teams and releases



Release Trains + Feature Toggles



Context



Discussion

- Benefits?
- Challenges?
- Implementation pitfalls?



Poll Everywhere Time!

Join by Web PollEv.com/potanim Join by Text Send **potanim** to **22333** 

What's your favourite aspect of Spotify Squad structure?  0

Squad **(A)**

Tribe **(B)**

Chapter **(C)**

Guild **(D)**

I didn't get it **(E)**



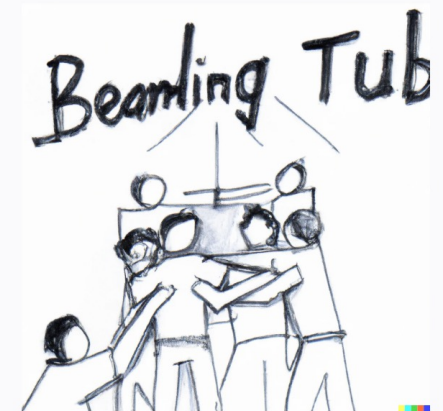
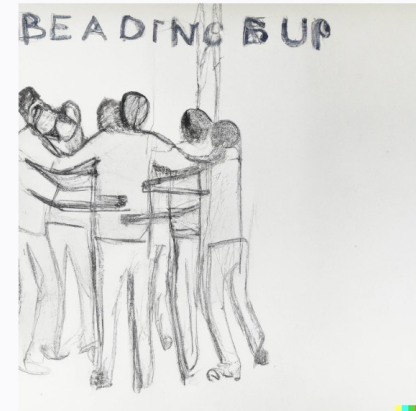
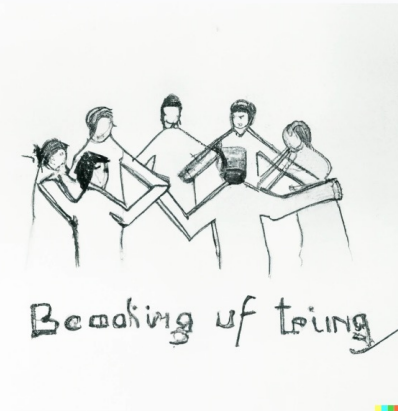
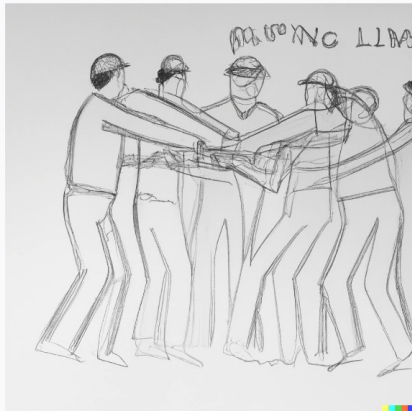


Edit the detailed description

Surprise me Upload →

pencil drawing of team building exercise

Generate

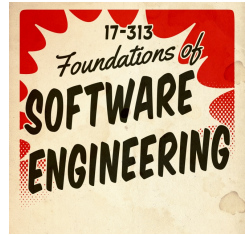


General Guidelines

Hints for team functioning

- Trust them; strategic not tactical direction
- Reduce bureaucracy, protect team
- Physical co-location, time for interaction
- Avoid in-team competition (bonuses etc)
- Time for quality assurance, cult of quality
- Realistic deadlines
- Peer coaching
- Sense of elitism
- Allow and encourage heterogeneity

DeMarco and Lister. Peopleware. Chapter 23



Team Fusion

- Forming, Storming, Norming, Performing
- Preserve existing teams, resist project mobility



Elitism Case Study: The Black Team



- Legendary team at IBM in the 1960s
- Group of talented ("slightly better") testers
 - Goal: Final testing of critical software before delivery
- Improvement over first year
- Formed team personality and energy
 - "adversary philosophy of testing"
 - Cultivated image of destroyers
 - Started to dress in black, crackled laughs, grew mustaches
- Team survived loss of original members

DeMarco and Lister. Peopleware. Chapter 22




Troubleshooting Teams




- Cynicism as warning sign
- Training to improve practices
- Getting to know each other; celebrate success; bonding over meals
- “A meeting without notes is a meeting that never happened”
- Local versus Remote Teams?
- Post-COVID Teams?



Poll Everywhere Time!


Join by Web PollEv.com/potanim Join by Text Send **potanim** to **22333** 

State the most important property of a good team: 

Join by Web
PollEv.com/potanim

Join by Text
Send **potanim** to **22333**

Join by QR code
Scan with your camera app





Edit the detailed description

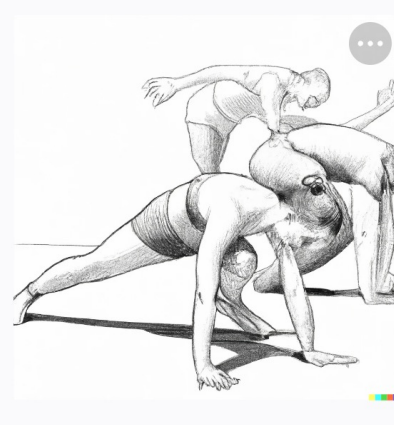
Surprise me

Upload



pencil drawing high turnover exercise

Generate



Developer Turnover and Motivation



Turnover



Rank	Employer Name	Median Age of Employees	Median Employee Tenure	Median Pay
1	Massachusetts Mutual Life Insurance Company	38	0.8	\$60,000
2 - tie	Amazon.com Inc	32	1.0	\$93,200
2 - tie	American Family Life Assurance Company of Columbus (AFLAC)	38	1.0	\$38,000
4 - tie	Google, Inc.	29	1.1	\$107,000
4 - tie	Mosaic	37	1.1	\$69,900
6 - tie	Chesapeake Energy Corporation	31	1.2	\$60,500
6 - tie	Group 1 Automotive, Inc.	32	1.2	\$33,200
6 - tie	Ross Stores, Inc	29	1.2	\$23,800
6 - tie	Wellcare Health Plans, Inc.	38	1.2	\$49,900
*				
11 - tie	Amerigroup Corporation	39	1.3	\$54,800
11 - tie	Brightpoint North America, Inc.	45	1.3	\$42,100
11 - tie	Devon Energy Corporation	31	1.3	\$63,200
11 - tie	Family Dollar Stores Inc	38	1.3	\$23,400
11 - tie	Freeport-McMoRan Copper & Gold Inc	36	1.3	\$62,900
11 - tie	Paccar Corporation	33	1.3	\$62,200
17	New York Life Insurance Company	33	1.4	\$53,800
18 - tie	Berkshire Hathaway Inc	41	1.5	\$53,600
18 - tie	Sandisk Corp	34	1.5	\$110,000
18 - tie	Tenneco Inc	40	1.5	\$69,900

Source: <http://www.techrepublic.com/blog/career-management/tech-companies-have-highest-turnover-rate/>; payscale.com data



Turnover

- > 20% turnover per year typical
 - average employment 15-36 month
- Costs?
- Reasons?
- Mitigations?



Unfolding Model of Employee Turnover



Organizational Science has studied employee turnover for over 100 years!

One Hundred Years of Employee Turnover Theory and Research

Peter W. Hom
Arizona State University

Thomas W. Lee
University of Washington

Jason D. Shaw
Hong Kong Polytechnic University

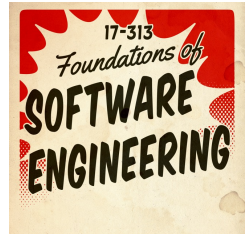
John P. Hausknecht
Cornell University

We review seminal publications on employee turnover during the 100-year existence of the *Journal of Applied Psychology*. Along with classic articles from this journal, we expand our review to include other publications that yielded key theoretical and methodological contributions to the turnover literature. We first describe how the earliest papers examined practical methods for turnover reduction or control and then explain how theory development and testing began in the mid-20th century and dominated the academic literature until the turn of the century. We then track 21st century interest in the psychology of staying (rather than leaving) and attitudinal trajectories in predicting turnover. Finally, we discuss the rising scholarship on collective turnover given the centrality of human capital flight to practitioners and to the field of human resource management strategy.

Keywords: embeddedness, employee turnover, job attitudes, shocks, participation mindsets



High turnover is expensive

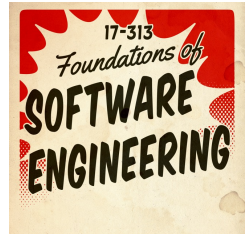


- Hiring overhead
 - Costs (1.5 month salary to agency)
 - Lost productivity (interviews)
- Getting new developers up to speed
 - Unproductive time (~6 month ramp up; 2 years in some estimates)
 - Training overhead
- Overhead for maintaining abandoned code
- Tendency to short-term viewpoints
- Premature promotions
- Young inexperienced staff

see also DeMarco and Lister. Peopleware. Chapter 19



Causes of, mitigations for turnover



- Causes (from literature, caveats for tech companies):
 - Just-passing-through mentality
 - Feeling of disposability
 - “Loyalty would be ludicrous”
 - High turnover encourages turnover
- Mitigations:
 - Environment and culture
 - striving to be "the best"
 - teams
 - Investment in personal growth, via retraining, no dead-end jobs
- *Advice: enable appropriate processes to maintain productivity despite turnover.*

see also DeMarco and Lister. Peopleware. Chapter 19





Growth and Challenge

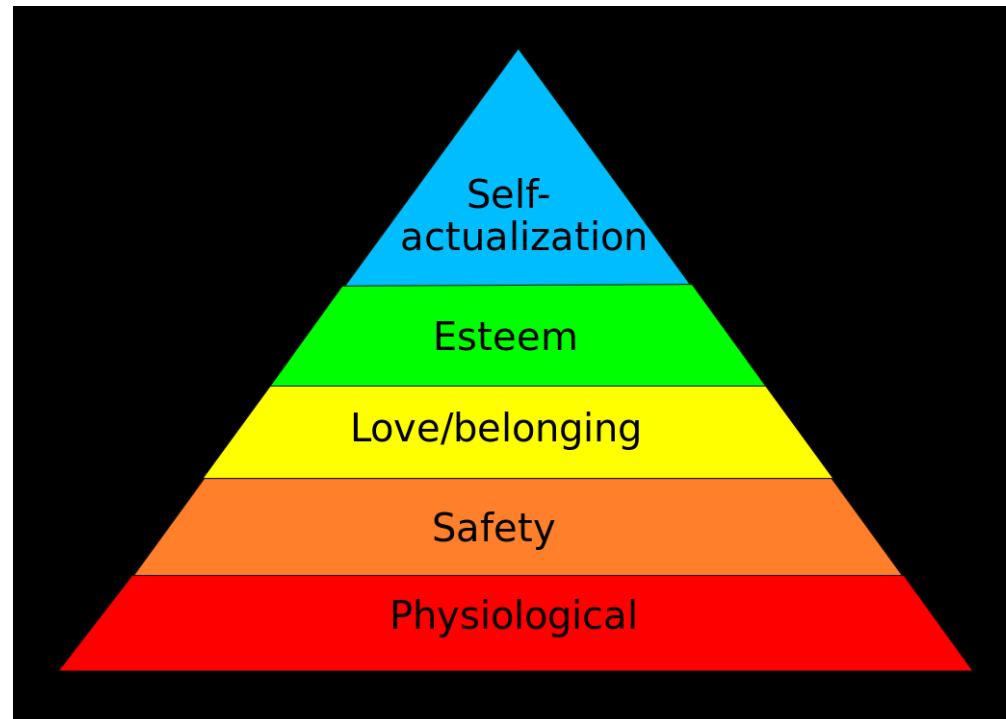


Theories

- Maslow's Hierarchy of Needs
- Herzberg's Motivation and Hygiene Factors
- Daniel Pink, Drive: The Surprising Truth About What Motivates Us.



Maslow's hierarchy of needs (1943)

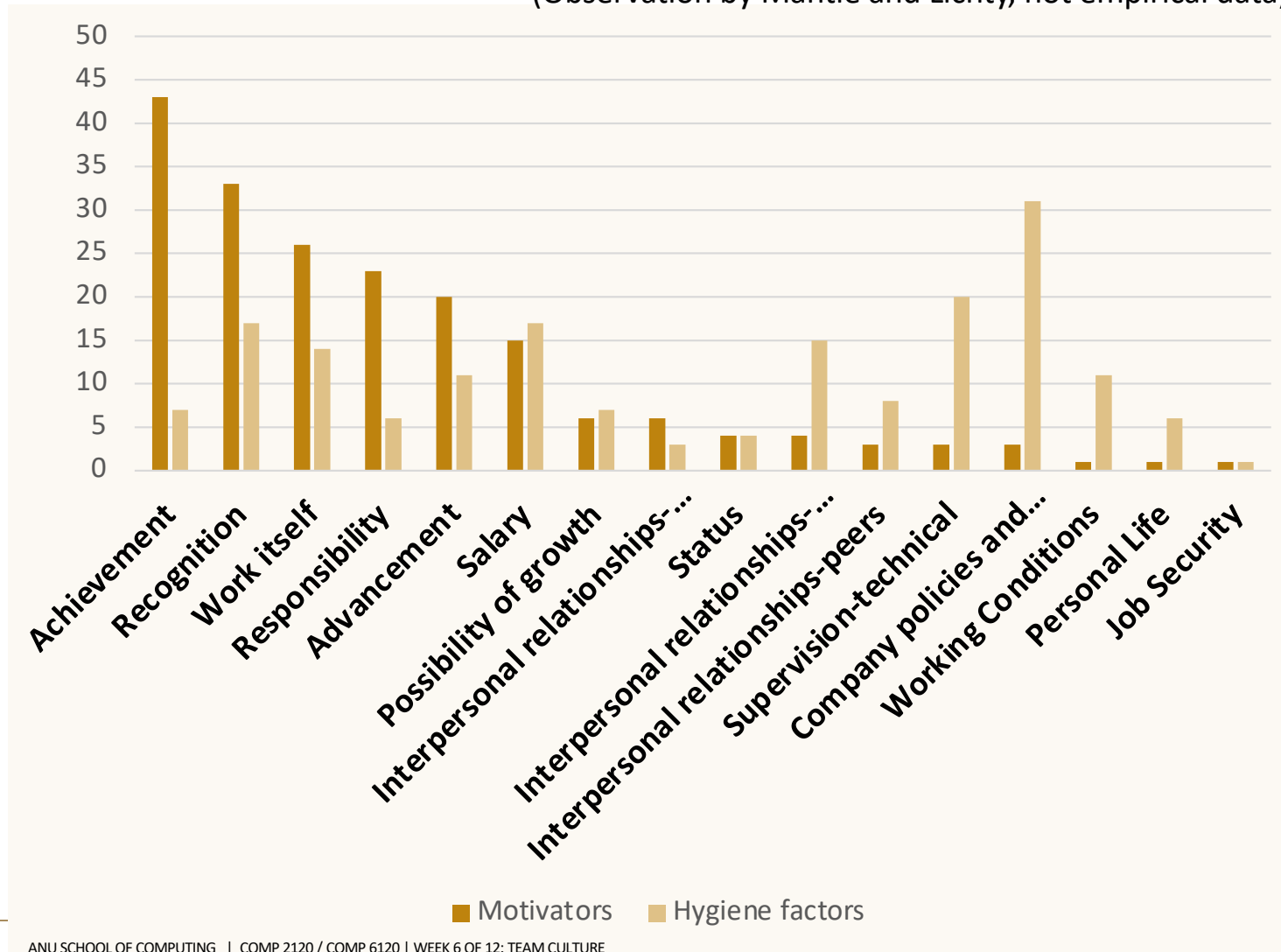


Herzberg's Motivation and Hygiene Factors (1960s)

- (aka two-factor theory)
- Different factors for satisfaction and dissatisfaction
 - Addressing dissatisfaction does not lead to satisfaction
- Step 1: Eliminate dissatisfaction
- Step 2: Create condition for satisfaction



(Observation by Mantle and Lichy, not empirical data)



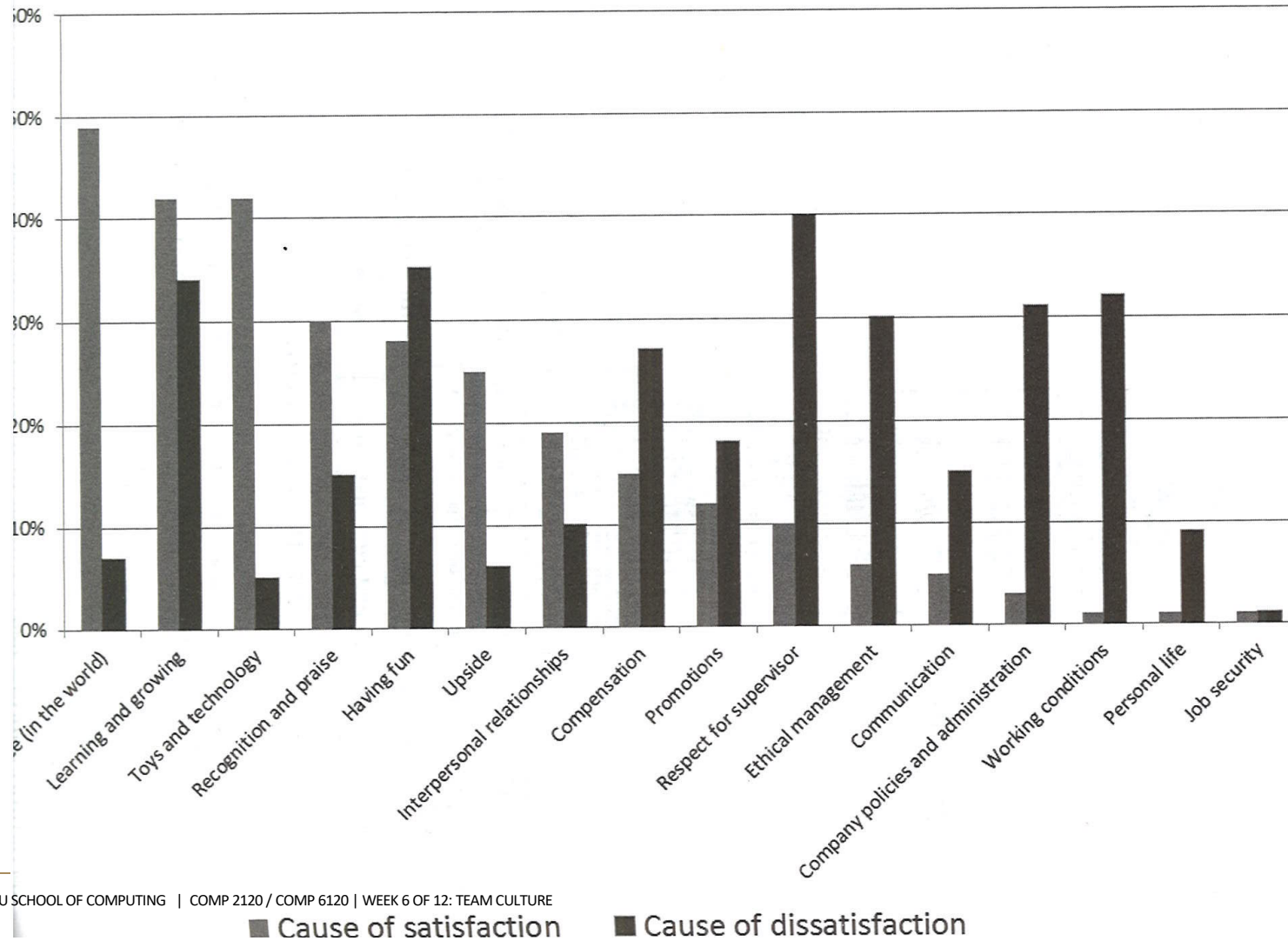
Identify Motivation and Hygiene Factors for Programmers



- Communication
- Company policies and administration
- Compensation
- Ethical management
- Having fun
- Interpersonal relationships
- Job security
- Learning and growing
- Making a difference
- Promotions
- Recognition and praise
- Respect for supervisor
- Toys and technology
- Upside



(Observation by Mantle and Lichty, not empirical data)



Addressing Causes of Dissatisfaction

- Respect for supervisor
- Having fun
- Learning and growing
- Good working conditions
- Sane company policies and administration
- Ethical management
- Fair compensation

- (often within control)



Addressing Causes of Dissatisfaction (selective)



- **Respect as supervisor**
 - gain technical credit
 - respect others
 - lead by example
 - help solve technical problems
 - manage and coach
- **Having fun**
 - out of office play
 - celebrations of accomplishments and occasions



Addressing Causes of Dissatisfaction (selective)



- Learning and growing
 - protect time for learning
 - explore new technologies; prototype
 - budget for attending conferences, seminars, in-house training
 - invite guest speakers
- Good working conditions
 - plenty of whiteboards
 - room for discussions
 - Quiet space, Limit interruptions, avoid meeting culture
 - cubicles vs separate offices
 - fire “jerks”
 - free food
 - flexible hours, flexible dress, flexible space



Addressing Causes of Dissatisfaction (selective)



- Sane company policies and administration
 - communicate frequently (vision, intentions, requirements, schedules, ...)
 - protect staff from organizational distractions
 - protect staff from bad communication practices (establish culture)



Addressing Motivating Factors (selective)



- Making a difference
 - worthy goals, long term vision
 - Steve Jobs when recruiting John Scully from Pepsi: “Do you want to sell sugar water or change to world”
- Toys and technology
 - modern hardware, large screens, phones, ...

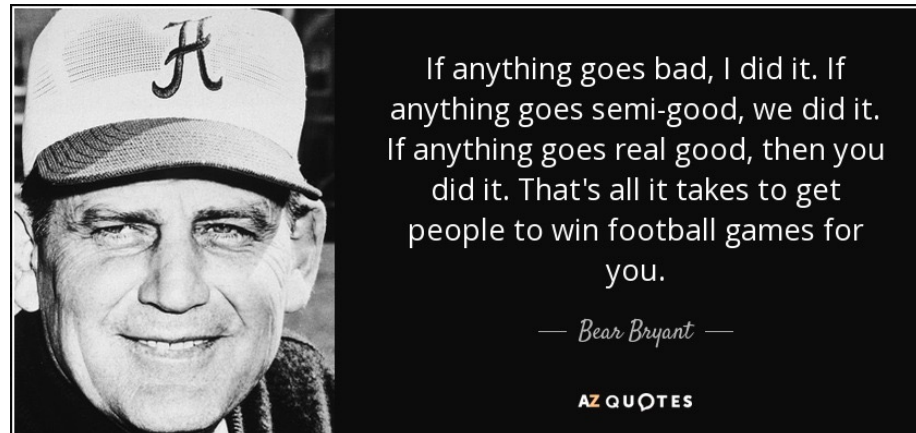


Addressing Motivating Factors (selective)



- Recognition and praise

- praise loudly and specifically, blame softly/privately
- celebrate success



Why do engineers choose TO JOIN particular teams?



Reasons grouped by clustering analysis	Percent
Liked new team and/or technology (exciting, manager)	85.8%
Coworker asked me to join (new team, old team)	37.8%
Joined for better opportunities (location, domain, lack of other options)	24.5%
Followed my manager (former or current)	14.6%

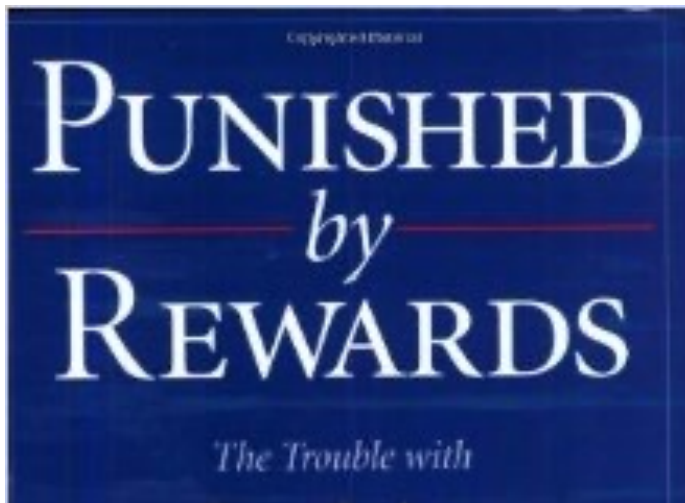


Why do engineers want to leave their teams?



Reasons grouped by clustering analysis	Percent
Change is coming (technology, charter, re-org, turnover)	52.6%
Seeking new challenges or location (role, location, challenges)	39.0%
Dissatisfaction with manager (priorities, goals, person, actions)	31.6%
The grass is always greener on the other side (novelty, escape)	12.3%
Not a good fit (bored, no need for my skills)	5.3%
Poor team dynamics (dysfunctional, no career growth)	4.4%





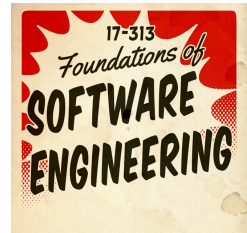
- Can extinguish intrinsic motivation
- Can diminish performance
- Can crush creativity
- Can crowd out good behavior
- Can encourage cheating, shortcuts, and unethical behavior
- Can become addictive
- Can foster short-term thinking

THE NEW YORK TIMES
'PROVOCATIVE' MA
'ENERGETIC' FINANCIAL TIMES
'INSPIRING' GUARDIAN

Rewards turn play into work and drain motivation

Autonomy
Mastery
Purpose

THE SURPRISING TRUTH ABOUT WHAT MOTIVATES US
DANIEL H. PINK



Rewards (aka grinding)



Free stones: +6,507
+22 per second

Scissors
Useful on the cut

Public Figures
Use their influence to help free stones

UC Kidney Stones
Your clicks free 2X more stones
CONVINCE (costs +117,649)

UC Stoners
Your clicks free 1.5X more stones
CONVINCE (costs +16,807)

41%





Rewards

- Fair and adequate pay as foundation
 - Takes money issue off the table, focus on work
- Non contingent awards
 - Reinforce extra effort and excellence if not expected



Daytime distractions



- “You never get anything done around here between 9 and 5”
- “I get my best work done in the early morning, before anybody else arrives”
- “In one late evening, I can do two or three days’ worth of work”

DeMarco and Lister. Peopleware. Chapter 8



Quality



- Quality important for satisfaction and self-image
- Even if market may not pay for that level or quality
- "Under time pressure developers sacrifice quality and hate themselves for that"
- "Quality, far beyond that required by the end user, is a means to higher productivity"

DeMarco and Lister. Peopleware. Chapter 4



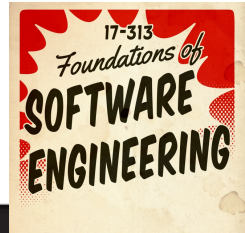
Overtime?

- Often caused by deadlines (real or artificial)
- Avoid over long periods
- Peak productivity at 40h/week
- Incentivize overtime
 - free food/overtime refrigerators
 - bonus pay
 - social pressure





Avoid “Gotcha Benefits”

- Fully paid vacations every year, including airfare
- Three-day weekends all summer.
- 30-day-paid sabbaticals every three years.
- \$1,000 per year continuing-education stipend. (learn anything)
- \$2,000 per year charity match.
- A local monthly CSA (community-supported agriculture) share
- One monthly massage at an actual spa, not the office.
- \$100 monthly fitness allowance



Poll Everywhere Time!


Join by Web PollEv.com/potantin Join by Text Send **potantin** to **22333** 

How many years do you think you will stay with your first full time employer after your graduation? 

Join by Web
PollEv.com/potantin

Join by Text
Send **potantin** to **22333**

Join by QR code
Scan with your camera app



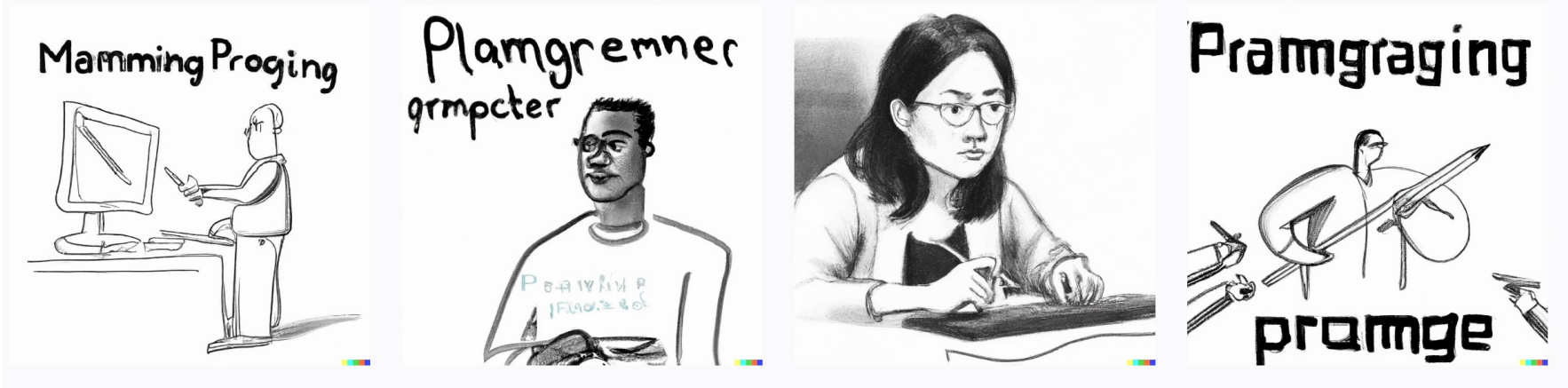


Edit the detailed description

Surprise me Upload →

pencil drawing of managing programmers

Generate



Managing Programmers

Programmer Characteristics



- Programmers have fun
 - Work is the primary motivator, not compensation
- Free spirited
 - Medium only “slightly removed from pure though stuff”
- Code right away, design as they go along
- Resistance to change
- Overconfident in own ability (writing bug free code, time estimation)



Managing Developers

- “Cowboys versus Farmers”
- Earn technical respect
- Protect staff from bureaucracy
- Set goals, avoid rewards



Tuckman, 1965: Forming, Storming, Norming, Performing



- **Forming:** team meets and learns about challenges, agrees on goals, begins to work.
 - Team members: (1) Behave independently. (2) May be motivated, but relatively uninformed about goals, (3) usually on their best behavior (albeit self-involved)
- **Storming:** participants form opinions about one another, possibly leading to conflict.
 - May voice opinions or question leader, especially if someone shirking responsibility or attempting to dominate.
 - Disagreements and conflicts must be resolved before team can progress; may regress if new challenges arise.
 - Stage can be destructive, but can lead to a better team in the long run if effective resolution tactics established.
- **Norming:** Resolved conflicts leads to a spirit of co-operation.
 - Team shares a common goal for which everyone takes responsibility.
 - Tolerate one another, move on from individual challenges.
 - Danger: too much avoidance of conflict can lead to avoidance of controversial ideas.
- **Performing:** group members focus on achieving common goals.
 - Everyone is now competent and can make decisions without supervision. Dissent is allowed if it's through acceptable channels.
 - Supervisors are almost always participating.
- **Upshot:** Preserve existing teams, resist project mobility.
 - Tradeoffs? Compared to practices you've seen in companies?




Further Reading




- Mantle and Lichty. *Managing the Unmanageable*. Addison-Wesley, 2013
 - Very accessible and practical tips at recruiting and management
- DeMarco and Lister. *Peopleware*. 3rd Edition. Addison Wesley, 2013
 - Anecdotes, stories, and tips on facilitating teams, projects, and environments
- Pink. *Drive: The Surprising Truth About What Motivates Us*. Riverhead 2011
 - Detailed discussion of motivating factors for creative people
- Sommerville. *Software Engineering*. 8th Edition. Chapter 25



Poll Everywhere Time!

Join by Web PollEv.com/potanim Join by Text Send **potanim** to **22333** 

Are you a Cowboy or a Farmer Programmer? 

Cowboy (skips planning, one-off solutions) **(A)**

Farmer (methodical, understand the environment first) **(B)**

I refuse to admit my secret personality **(C)**





Edit the detailed description

Surprise me

Upload →

pencil drawing of software documentation

Generate



Documentation

Documentation (Chapter 10 of SE @ Google)

- **What Qualifies as Documentation?**
 - Any supplemental text that an engineer needs to write to do their job (including comments).
- **Why is Documentation Needed?**
 - Helps formulate an API (writing docs helps figure out if it makes sense)
 - Provides roadmap for maintenance and history
 - Makes code look more professional and attractive
 - Prompts fewer questions from other users

“optimise for the reader”



Documentation is Like Code

- Your documentation *should*:
 - Have internal policies or rules to be followed
 - Be placed under source control
 - Have clear ownership responsible for maintaining the docs
 - Undergo reviews for changes (and change *with* the code it documents)
 - Have issues tracked, as bugs are tracked in code
 - Be periodically evaluated (tested, in some respect)
 - If possible, be measured for aspects such as accuracy, freshness, etc. (need more tools here!)



Case Study: The Google Wiki

When Google was much smaller and leaner, it had few technical writers. The easiest way to share information was through our own internal wiki (GooWiki). At first, this seemed like a reasonable approach; all engineers shared a single documentation set and could update it as needed.

But as Google scaled, problems with a wiki-style approach became apparent. Because there were no true owners for documents, many became obsolete.³ Because no process was put in place for adding new documents, duplicate documents and document sets began appearing. GooWiki had a flat namespace, and people were not good at applying any hierarchy to the documentation sets. At one point, there were 7 to 10 documents (depending on how you counted them) on setting up Borg, our production compute environment, only a few of which seemed to be maintained, and most were specific to certain teams with certain permissions and assumptions.

Another problem with GooWiki became apparent over time: the people who could fix the documents were not the people who used them. New users discovering bad documents either couldn't confirm that the documents were wrong or didn't have an easy way to report errors. They knew something was wrong (because the document didn't work), but they couldn't "fix" it. Conversely, the people best able to fix the documents often didn't need to consult them after they were written. The documentation became so poor as Google grew that the quality of documentation became Google's number one developer complaint on our annual developer surveys.

The way to improve the situation was to move important documentation under the same sort of source control that was being used to track code changes. Documents began to have their own owners, canonical locations within the source tree, and processes for identifying bugs and fixing them; the documentation began to dramatically improve. Additionally, the way documentation was written and maintained began to look the same as how code was written and maintained. Errors in the documents could be reported within our bug tracking software. Changes to the documents could be handled using the existing code review process. Eventually, engineers began to fix the documents themselves or send changes to technical writers (who were often the owners).



Moving documentation to source control was initially met with a lot of controversy. Many engineers were convinced that doing away with the GooWiki, that bastion of freedom of information, would lead to poor quality because the bar for documentation (requiring a review, requiring owners for documents, etc.) would be higher. But that wasn't the case. The documents became better.

The introduction of Markdown as a common documentation formatting language also helped because it made it easier for engineers to understand how to edit documents without needing specialized expertise in HTML or CSS. Google eventually introduced its own framework for embedding documentation within code: **g3doc**. With that framework, documentation improved further, as documents existed side by side with the source code within the engineer's development environment. Now, engineers could update the code and its associated documentation in the same change (a practice for which we're still trying to improve adoption).

The key difference was that maintaining documentation became a similar experience to maintaining code: engineers filed bugs, made changes to documents in changelists, sent changes to reviews by experts, and so on. Leveraging of existing developer workflows, rather than creating new ones, was a key benefit.



Know Your Audience

Types of Audiences

- Experience level
- Domain knowledge
- Purpose
- *Seekers versus Stumblers*
- *Customer versus Provider*

Documentation Types

- Reference documentation (including comments)
- Design documents
- Tutorials
- Conceptual documentation
- Landing pages



Reference Documentation

- File Comments

```
// -----  
// str_cat.h  
// -----  
//  
// This header file contains functions for efficiently concatenating and appending  
// strings: StrCat() and StrAppend(). Most of the work within these routines is  
// actually handled through use of a special AlphaNum type, which was designed  
// to be used as a parameter type that efficiently manages conversion to  
// strings and avoids copies in the above operations.  
...
```

- Class Comments

```
// -----  
// AlphaNum  
// -----  
//  
// The AlphaNum class acts as the main parameter type for StrCat() and  
// StrAppend(), providing efficient conversion of numeric, boolean, and  
// hexadecimal values (through the Hex type) into strings.
```

- Function Comments

```
// Creates a new record for a customer with the given name and address,  
// and returns the record ID, or throws `DuplicateEntryError` if a  
// record with that name already exists.  
int AddCustomer(string name, string address);
```



Tutorials

Example: A bad tutorial

1. Download the package from our server at <http://example.com>
2. Copy the shell script to your home directory
3. Execute the shell script
4. The foobar system will communicate with the authentication system
5. Once authenticated, foobar will bootstrap a new database named “baz”
6. Test “baz” by executing a SQL command on the command line
7. Type: `CREATE DATABASE my_foobar_db;`

Example: A bad tutorial made better

1. Download the package from our server at <http://example.com>:

```
$ curl -I http://example.com
```

2. Copy the shell script to your home directory:

```
$ cp foobar.sh ~
```

3. Execute the shell script in your home directory:

```
$ cd ~; foobar.sh
```

The foobar system will first communicate with the authentication system. Once authenticated, foobar will bootstrap a new database named “baz” and open an input shell.

4. Test “baz” by executing a SQL command on the command line:

```
baz:$ CREATE DATABASE my_foobar_db;
```

Note how each step requires specific user intervention. If, instead, the tutorial had a focus on some other aspect (e.g., a document about the “life of a server”), number those steps from the perspective of that focus (what the server does).





Documentation Reviews

- Technical Review (accuracy)
- Audience Review (clarity)
- Writing Review (consistency)
- Documentation Philosophy:
 - ~~HOW~~
 - WHO (audience)
 - WHAT (purpose of the doc)
 - WHEN (created/reviewed/updated)
 - WHERE (*ideally with source code it documents*)
 - WHY (what to take away after reading)



Poll Everywhere Time!

Join by Web Pollev.com/potantin Join by Text Send **potantin** to **22333** 

Percentage of my time spent on writing documentation when coding a project is:  0

0-10% **(A)**

10-25% **(B)**

25-50% **(C)**

50-75% **(D)**

75-100% (when do you code then???) **(E)**

