Foundations of Software Engineering

Part 24: Teams
Christian Kästner
Case Studies

Disclaimer: All pictures represent abstract developer groups or products to give a sense of scale; they are not necessarily the developers of those products or developers at all.
How to structure teams?

• Microblogging platform; 3 friends
How to structure teams?

- Banking app; 15 developers
How to structure teams?

• Mobile game; 50ish developers;
• distributed teams?
How to structure teams?

• Mobile game; 200ish developers
How to structure teams?

• Ride sharing app and self-driving cars; 1200 developers; 4 sites
Teams
Necessity of Groups

• Division of labor
• Division of expertise (e.g., security expert, database expert)
Team Issues

- Process costs
- Groupthink
- Social loafing
- Multiple/conflicting goals
Team issues: Process costs
Mythical Man Month

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

1975, describing experience at IBM developing OS/360
Process Costs

\[ n(n - 1) / 2 \]

communication links
Process Costs
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.
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)
• 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
Agile Practices (e.g., Scrum)

• 7+/-2 team members, collocated
• self managing
• Scrum master (potentially shared among 2-3 teams)
• Product owner / customer representative
Mantle and Lichty

• Ideal team size: 2-3 colocated developers if possible
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 mount 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
Team issues: Groupthink
WE NEED MORE DISSenting OpINIONS.

WE AGREE 100%
Groupthink

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


Unconscious bias

• Pervasive, cultural
• Raise awareness
• Explicit goals
• Measurement
**FastLane** is an interactive real-time system used to conduct NSF business over the Internet. FastLane is for official NSF use only. More About FastLane...

---

**Quick Links**

- Help for Proposal Preparation
- Frequently Asked Questions About FastLane Proposal Preparation
- Grant Proposal Guide
- Deadlines and Target Dates
- Change Password
- Lookup NSF ID

---

**Proposals, Awards and Status**

Log in for the following permission-based functions:

- **Proposal Functions**
  - Letters of Intent
  - Proposal Preparation
  - Proposal Status
  - Display Reference Status
  - Revise Submitted Proposal Budget
  - Proposal File Update

- **Award and Reporting Functions**
  - Notifications and Requests - Disabled in FastLane. Log in to Research.gov
  - Continuation Funding Status
  - View/Print Award Documents
  - Project Reports System - Disabled in FastLane. Log in to Research.gov
  - Supplemental Funding Request

- Change PI Information

---

**PI/Co-PI Log In**

- Last Name: Kästner
- NSF ID:
- Privacy Act: ••••
- Password: ••••

Log In

Forgot Password?
Lookup NSF ID

---

**Other Authorized Users (OAU) Log In**

Log In by Proposal ID

- OAU Last Name:
- OAU NSF ID:

---
Mitigation Strategies

• Several agile techniques
  – 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 issues: Social loafing
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”

Mitigation Strategies

• Involve all team members, colocation
• 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
Agile Practices as Mitigation?
Responsibilities & Buy-In

• Involve team members in decision making
• Assign responsibilities (ideally goals not tasks)
• Record decisions and commitments; make record available
Autonomy
Mastery
Purpose
Team issues: Multiple/conflicting goals
How can I Make it?

How can I Break it?
Incentives?

• Team incentives
• vs individual incentives?
Agile Practices as Mitigation?
Matrix Organization

Temporary assignment to projects; flexible staffing
Project Organization

mgmt

System programmers
Application programmers
QA
Security
Marketing

Project 1
Project 2
Project 3
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.

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

• Many failures

• Isolated, but very significant, success
  — Jazz, Github, ...
General Guidelines
Hints for team functioning

• Trust them; strategic not tactical direction
• Reduce bureaucracy, protect team
• Physical colocation, 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
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
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”
Further Reading

• Mantle and Lichty. Managing the Unmanageable. Addison-Wesley, 2013
  – Very accessible and practical tips at recruiting and management

  – Anecdotes, stories, and tips on facilitating teams, projects, and environments

• Sommerville. Software Engineering. 8th Edition. Chapter 25