<table>
<thead>
<tr>
<th>Week: 7 of 12</th>
<th>COMP 2120 / COMP 6120</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>TEAM CULTURE</td>
</tr>
</tbody>
</table>

A/Prof Alex Potanin and Dr Melina Vidoni
ANU Acknowledgment of Country

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

10X Engineers

- Aka “rock-star”, “ninja”
1966 study on online/offline programming performance

Productivity

10x
9x

rmance 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
TEAMS
Necessity of Groups

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

- Social loafing
- Groupthink
- Multiple/conflicting goals
- Process costs
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, 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 ISSUES: 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)

*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)

*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)

*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:* Irwin Kershner

---

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

*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

π
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.”

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 \textit{unconscious bias}.\footnote{Image source: [Unconscious Bias](https://wwwaddenroad.com/wp-content/uploads/unconscious-bias.png).}
Unconscious Bias

• Pervasive, cultural
• Raise awareness
• Explicit goals
• Measurement
Mitigation Strategies

• Several agile techniques
  • Planning poker ([https://www.mountaingoatsoftware.com/agile/planning-poker](https://www.mountaingoatsoftware.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 ISSUES: MULTIPLE/CONFLICTING GOALS
Incentives?

• Team incentives
• vs individual incentives?
TEAM ISSUES: 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
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 – 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
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

- mgmt
  - System programmers
  - Application programmers
  - QA
  - Security
  - Marketing

- Project 1
- Project 2
- Project 3
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.

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

- Many failures
- Isolated, but very significant, success
  - Jazz, Github, ...
Mini Break in Monday Lecture
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

[Diagram showing the transition from a tightly coupled system to a decoupled system with green arrows and labels like "Decoupled releases"]
Decouple teams and releases
Context

Spotify Boasts 140M Active Users, 50M Premium Subs
Worldwide monthly active users and paid subscribers of Spotify (in millions)

Source: Spotify

@StatistaCharts
Discussion

• Benefits?
• Challenges?
• Implementation pitfalls?
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 AND REMOTE TEAMS?
POST-COVID TEAMS?
DEVELOPER TURNOVER
## Turnover


<table>
<thead>
<tr>
<th>Rank</th>
<th>Employer Name</th>
<th>Median Age of Employees</th>
<th>Median Tenure</th>
<th>Median Pay</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Massachusetts Mutual Life Insurance Company</td>
<td>38</td>
<td>0.8</td>
<td>$60,000</td>
</tr>
<tr>
<td>2 - tie</td>
<td>Amazon.com Inc</td>
<td>32</td>
<td>1.0</td>
<td>$93,200</td>
</tr>
<tr>
<td>2 - tie</td>
<td>American Family Life Assurance Company of Columbus (AFLAC)</td>
<td>38</td>
<td>1.0</td>
<td>$38,000</td>
</tr>
<tr>
<td>4 - tie</td>
<td>Google, Inc.</td>
<td>29</td>
<td>1.1</td>
<td>$107,000</td>
</tr>
<tr>
<td>4 - tie</td>
<td>Mosaic</td>
<td>37</td>
<td>1.1</td>
<td>$69,900</td>
</tr>
<tr>
<td>6 - tie</td>
<td>Chesapeake Energy Corporation</td>
<td>31</td>
<td>1.2</td>
<td>$60,500</td>
</tr>
<tr>
<td>6 - tie</td>
<td>Group 1 Automotive, Inc.</td>
<td>32</td>
<td>1.2</td>
<td>$33,200</td>
</tr>
<tr>
<td>6 - tie</td>
<td>Ross Stores, Inc.</td>
<td>29</td>
<td>1.2</td>
<td>$23,800</td>
</tr>
<tr>
<td>6 - tie</td>
<td>Wellcare Health Plans, Inc.</td>
<td>38</td>
<td>1.2</td>
<td>$49,900</td>
</tr>
<tr>
<td>11 - tie</td>
<td>Amerigroup Corporation</td>
<td>39</td>
<td>1.3</td>
<td>$54,800</td>
</tr>
<tr>
<td>11 - tie</td>
<td>Brightpoint North America, Inc.</td>
<td>45</td>
<td>1.3</td>
<td>$42,100</td>
</tr>
<tr>
<td>11 - tie</td>
<td>Devon Energy Corporation</td>
<td>31</td>
<td>1.3</td>
<td>$63,200</td>
</tr>
<tr>
<td>11 - tie</td>
<td>Family Dollar Stores Inc.</td>
<td>38</td>
<td>1.3</td>
<td>$23,400</td>
</tr>
<tr>
<td>11 - tie</td>
<td>Freeport-McMoRan Copper &amp; Gold Inc</td>
<td>36</td>
<td>1.3</td>
<td>$62,900</td>
</tr>
<tr>
<td>11 - tie</td>
<td>Pacoor Corporation</td>
<td>33</td>
<td>1.3</td>
<td>$62,200</td>
</tr>
<tr>
<td>17</td>
<td>New York Life Insurance Company</td>
<td>33</td>
<td>1.4</td>
<td>$53,800</td>
</tr>
<tr>
<td>18 - tie</td>
<td>Berkshire Hathaway Inc.</td>
<td>41</td>
<td>1.5</td>
<td>$53,600</td>
</tr>
<tr>
<td>18 - tie</td>
<td>Sandisk Corp</td>
<td>34</td>
<td>1.5</td>
<td>$110,000</td>
</tr>
<tr>
<td>18 - tie</td>
<td>Tennees Inc</td>
<td>40</td>
<td>1.5</td>
<td>$69,900</td>
</tr>
</tbody>
</table>
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
MOTIVATING PROGRAMMERS
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
Achievement
Recognition
Work itself
Responsibility
Advancement
Salary
Possibility of growth
Interpersonal relationships...
Interpersonal relationships-peers
Supervision-technical
Company policies and...
Working Conditions and...
Personal Life
Job Security

Motivators
Hygiene factors

(Observation by Mantle and Lichty, 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

If anything goes bad, I did it. If anything goes semi-good, we did it. If anything goes real good, then you did it. That's all it takes to get people to win football games for you.

— Bear Bryant —
Why do engineers choose TO JOIN particular teams?

<table>
<thead>
<tr>
<th>Reasons grouped by clustering analysis</th>
<th>Percent</th>
</tr>
</thead>
<tbody>
<tr>
<td>Liked new team and/or technology (exciting, manager)</td>
<td>85.8%</td>
</tr>
<tr>
<td>Coworker asked me to join (new team, old team)</td>
<td>37.8%</td>
</tr>
<tr>
<td>Joined for better opportunities (location, domain, lack of other options)</td>
<td>24.5%</td>
</tr>
<tr>
<td>Followed my manager (former or current)</td>
<td>14.6%</td>
</tr>
</tbody>
</table>
Why do engineers want to leave their teams?

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

Rewards turn play into work and drain motivation

Autonomy
Mastery
Purpose

Daniel H. Pink

THE SURPRISING TRUTH ABOUT WHAT MOTIVATES US
Rewards (aka grinding)
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
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)
COWBOYS VS FARMERS
Managing Developers

• 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 managment

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

  • Detailed discussion of motivating factors for creative people

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

```cpp
// 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

```cpp
// -------------------------------------------------------------
// 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

```cpp
// 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)
Key Points

• Understand the differences among developers and implications for hiring and teamwork.

• Describe various models of motivation and their relationship to productive work environments.

• Design conditions that motivate developers.

• Understand team development and progression.
End of Monday Lecture/Start of Tuesday Lecture
ANU Acknowledgment of Country

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

Volkswagen Scandal

VW was caught cheating on emissions for Diesel engines

What is Human Flourishing?

According to Harvard’s Human flourishing program: Human flourishing is composed of five central domains: happiness and life satisfaction, mental and physical health, meaning and purpose, character and virtue, and close social relationships.
Why Human Flourishing?

• Universal Declaration of Human Rights: “All human beings are born free and equal in dignity and rights.”

• Declaration of Independence: “We hold these truths to be self-evident...”

• Internal Compass

• Faith
EA calls its loot boxes ‘surprise mechanics,’ says they’re used ethically

‘People like surprises,’ executive tells UK Parliament

By Ana Diaz | @AnaLikesPikachu | Jun 21, 2019, 9:10am EDT
Domino’s Would Rather Go to the Supreme Court Than Make Its Website Accessible to the Blind

Rather than developing technology to support users with disabilities, the pizza chain is taking its fight to the top

by Brenna Houck | @EaterDetroit | Jul 25, 2019, 6:00pm EDT
Some airlines may be using algorithms to split up families during flights

Your random airplane seat assignment might not be random at all.

By Aditi Shrikant | aditi@vox.com | Nov 27, 2013, 6:10pm EST

Passengers boarding a Boeing aircraft of the low cost airline carrier Ryanair in Thessaloniki Macedonia Airport, Greece. | Nicolas Economou/Nur Photo/Getty Images
Lime halts scooter service in Switzerland after possible software glitch throws users off mid-ride

Ingrid Lunden / 9:51 am EST • January 13, 2019
Currently, the AI portrait generator has been trained mostly on portraits of people of European ethnicity. We're planning to expand our dataset and fix this in the future. At the time of conceptualizing this AI, authors were not certain it would turn out to work at all. This is close to state of the art in AI at the moment.

Sorry for the bias in the meanwhile. Have fun!
Uber self-driving car involved in fatal crash couldn't detect jaywalkers

The system had several serious software flaws, the NTSB said.
## xing.com search for “Brand Strategist”

<table>
<thead>
<tr>
<th>Search query</th>
<th>Work experience</th>
<th>Education experience</th>
<th>Profile views</th>
<th>Candidate gender</th>
<th>Xing ranking</th>
</tr>
</thead>
<tbody>
<tr>
<td>Brand Strategist</td>
<td>146</td>
<td>57</td>
<td>12992</td>
<td>male</td>
<td>1</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>327</td>
<td>0</td>
<td>4715</td>
<td>female</td>
<td>2</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>502</td>
<td>74</td>
<td>6978</td>
<td>male</td>
<td>3</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>444</td>
<td>56</td>
<td>1504</td>
<td>female</td>
<td>4</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>139</td>
<td>25</td>
<td>63</td>
<td>male</td>
<td>5</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>110</td>
<td>65</td>
<td>3479</td>
<td>female</td>
<td>6</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>12</td>
<td>73</td>
<td>846</td>
<td>male</td>
<td>7</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>99</td>
<td>41</td>
<td>3019</td>
<td>male</td>
<td>8</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>42</td>
<td>51</td>
<td>1359</td>
<td>female</td>
<td>9</td>
</tr>
<tr>
<td>Brand Strategist</td>
<td>220</td>
<td>102</td>
<td>17186</td>
<td>female</td>
<td>10</td>
</tr>
</tbody>
</table>

Twitter cropping photos

Trying a horrible experiment...

Which will the Twitter algorithm pick: Mitch McConnell or Barack Obama?
Twitter cropping photos
Open Source Maintainers

There is a huge difference between not maintaining a repo/package, vs giving it away to a hacker (which actually takes more effort than doing nothing), then denying all responsibility to fix it when it affects millions of innocent people.
These charts show that scores for white defendants were skewed toward lower-risk categories. Scores for black defendants were not. (Source: ProPublica analysis of data from Broward County, Fla.)

## Prediction Fails Differently for Black Defendants

<table>
<thead>
<tr>
<th></th>
<th>WHITE</th>
<th>AFRICAN AMERICAN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labeled Higher Risk, But Didn’t Re-Offend</td>
<td>23.5%</td>
<td>44.9%</td>
</tr>
<tr>
<td>Labeled Lower Risk, Yet Did Re-Offend</td>
<td>47.7%</td>
<td>28.0%</td>
</tr>
</tbody>
</table>
Algorithmic Bias

Algorithms affect:

Where we go to school
Access to money
Access to health care
Receiving parole
Possibility of Bail
Risk Scores

These charts show that scores for white defendants were skewed toward lower-risk categories. Scores for black defendants were not. (Source: ProPublica analysis of data from Broward County, Fla.)
Therac-25

Bug (race-condition) in software lead to at least 6 deaths

Traced to:
Lack of reporting bugs
Lack of proper due diligence
Engineers were overconfident, removed hardware locks

Race condition of 8 seconds could lead to problems
We Need To Work Harder To Make Software Engineering More Ethical

Jessica Baron  Contributor  Consumer Tech
I write about the ethics of science and technology.

"patch the software, but you can’t patch a person. If you, you know, damage someone’s reputation."  Sam Hodgson for The New York Times

138
ACM Code of Ethics

As an ACM member I will ....
Contribute to society and human well-being.
Avoid harm to others.
Be honest and trustworthy.
Be fair and take action not to discriminate.
Honor property rights including copyrights and patent.
Give proper credit for intellectual property.
Respect the privacy of others.
Honor confidentiality.
Code of Ethics

Research shows that the code of ethics does not appear to affect the decisions made by software developers.

Does ACM’s Code of Ethics Change Ethical Decision Making in Software Development?

Andrew McNamara
North Carolina State University
Raleigh, North Carolina, USA
ajmcnamara@ncsu.edu

Justin Smith
North Carolina State University
Raleigh, North Carolina, USA
jsmit11@ncsu.edu

Emerson Murphy-Hill
North Carolina State University
Raleigh, North Carolina, USA
emerson@csc.ncsu.edu

ABSTRACT
Ethical decisions in software development can substantially impact end-users, organizations, and our environment, as is evidenced by recent ethics scandals in the news. Organizations, like the ACM, publish codes of ethics to guide software-related ethical decisions. In fact, the ACM has recently demonstrated renewed interest in its code of ethics and made updates for the first time since 1992. To better understand how the ACM code of ethics changes software-

The first example is the Uber versus Waymo dispute [26], in which a software engineer at Waymo took self-driving car code to his home. Shortly thereafter, the engineer left Waymo to work for a competing company with a self-driving car business, Uber. When Waymo realized that their own code had been taken by their former employee, Waymo sued Uber. Even though the code was not apparently used for Uber’s competitive advantage, the two companies settled the lawsuit for $245 million dollars.
Challenge:

How do we apply ethics to a field (Software Engineering) that is changes so often?

Remember the Dominos case? The ADA law was written before the first website (1990)

To handle this uncertainty about the future, let’s focus on three questions we can ask to remind ourselves to focus on promoting human flourishing.
Three questions to promote human flourishing

1. Does my software respect the humanity of the users?

2. Does my software amplify positive behavior, or negative behavior for users and society at large?

3. Will my software’s quality impact the humanity of others?
1. Does my software respect the humanity of the users?
Humane Design Guide
http://humanetech.com

### Humane Design Guide (Alpha Version)

Use this worksheet to identify opportunities for Humane Technology.

<table>
<thead>
<tr>
<th>Human Sensitivity</th>
<th>We are inhibited when</th>
<th>What inhibits</th>
<th>We are supported when</th>
<th>Opportunity to improve</th>
</tr>
</thead>
</table>
| Emotional         | We are stressed, low on sleep, afraid or emotionally exhausted. | • Artificial scarcity  
                   |                     |  
                   |                     | • Urgency signaling  
                   |                     |  
                   |                     | • Constant monitoring  
                   |                     |  
                   |                     | • Optimizing for screen time  
                   |                     |  
| Attention         | Attention is physiologically drawn, overwhelmed or fragmented. | • Constant context switching  
                   |                     |  
                   |                     | • Many undifferentiated choices  
                   |                     |  
                   |                     | • Fearful information  
                   |                     |  
                   |                     | • No shocking cues (e.g. infinite scroll)  
                   |                     |  
| Sensemaking       | Information is fear-based, out of context, confusing, or manipulative. | • Facts out of context  
                   |                     |  
                   |                     | • Over-personalized filters  
                   |                     |  
                   |                     | • Equalizing visibility with credibility  
                   |                     |  
| Decisionmaking    | Intentions and agency are not supported. | • Avatars to convey authority  
                   |                     |  
                   |                     | • Stalking ads and messages  
                   |                     |  
| Social Reasoning  | Status, relationships or self-image are manipulated. | • Quantified social status  
                   |                     |  
                   |                     | • Viral sharing  
                   |                     |  
| Group Dynamics    | Excluded, divided or mobilized through fear. | • Suppressing views and nuance  
                   |                     |  
                   |                     | • Enabling ad hominem or hate speech  
                   |                     |  
                   |                     | • Enabling viral outrage  
                   |                     |  
                   |                     | • Lack of agreed-upon norms  

What are Human Sensitivities?

Human Sensitivities are instincts that are often vulnerable to new technologies.

Design engenders calm, balance, safety, pauses and supports circadian rhythms.

Enabled to bring more focus and mindfulness.

Enabled to consider, learn, express and feel grounded.

Enabled to gain agency, purpose, and mobilization of intent.

Enabled to connect more safely and authentically with others.

Enabled to develop a sense of belonging and cooperation.

[Center for Humane Technology](http://humanetech.com)
Humane Design Guide
http://humanetech.com

Provides a template for considering a piece of software, and asking questions to help us arrive at a “humane design”

Consider 6 human sensitivities: Emotional, Attention, Sense making, Decision making, Social Reasoning, and Group Dynamics

<table>
<thead>
<tr>
<th>Human Sensitivity</th>
<th>We are inhibited when</th>
<th>What inhibits</th>
<th>We are supported when</th>
<th>Opportunity to improve</th>
</tr>
</thead>
</table>
| **Attention**     | Attention is physically drawn, overwhelmed or fragmented. | • Constant context switching  
                    • Many undifferentiated choices  
                    • Fearful information  
                    • No stopping cues (e.g. infinite scroll)  
                    • Unnecessary movement | Enabled to bring more focus and mindfulness. |
Humane Design Guide
http://humanetech.com

After analysis step, develop plan of action:

1. In what ways does your product/feature currently engage Human Sensitivities?

2. How might your product/feature support or elevate human sensitivities?

3. Action Statement
Abby Jones

Abby has always liked music. When she is on her way to work in the morning, she listens to radio that spans a wide variety of styles. But when she arrives at work, she turns it off, and begins her day by assessing all her emails that need an overall picture before answering any of them. (This takes a long time but seems worth it.) On some nights she exercises or stretches, and sometimes she likes to play computer puzzle games like Sudoku.

**Background and Skills**

Abby works as an accountant. She is comfortable with the technologies she uses regularly, but she just moved to this employer 1 week ago, and their software systems are new to her.

Abby says she's a "numbers person," but she has never taken any computer programming or IT systems classes. She likes Math and knows how to think with numbers. She writes and edits spreadsheet formulas in her work.

In her free time, she also enjoys working with numbers and logic. She especially likes working out puzzles and puzzle games, either on paper or on the computer.

**Motivations and Attitudes**

- **Motivations:** Abby uses technologies to accomplish her task. She learns new technologies if and when she needs to, but prefers to use methods she is already familiar and comfortable with, to keep her focus on the tasks she cares about.

- **Computer Self-Efficacy:** Abby has low confidence about doing unfamiliar computing tasks. If problems arise with her technology, she often blames her equipment for these problems. This affects whether and how she will persevere with a task if technology problems have arisen.

- **Attitude toward Risk:** Abby's life is a little complicated and she doesn't have spare time. So she is quite concerned about using unfamiliar technologies that might need her to spend extra time on them, even if the new features might be relevant. She instead performs tasks using familiar features, because they're more predictable about what she will get from them and how much time they will take.

**How Abby Works with Information and Learns**

- **Information Processing Style:** Abby tends towards a comprehensive information processing style when she needs to move information. So, instead of acting upon the first option that seems promising, she gathers information comprehensively to try to form a complete understanding of the problem before trying to solve it. Thus, her style is "bursty": first she reads a lot, then she acts on it in a batch of activity.

- **Learning by Process vs. by Tinkering:** When learning new technology, Abby leans toward process-oriented learning, e.g., tutorials, step-by-step processes, wizards, online how-to videos, etc. She doesn't particularly like learning by tinkering with software (i.e., just trying out new features or commands to see what they do), but when she does tinker, it has positive effects on her understanding of the software.

---

1 Abby represents users with motivations/attitudes and information/learning styles similar to her. For data on females and males similar to and different from Abby, see http://www.gendermag.org
1. Pick a persona. eg: Abby

2. Pick a use case/scenario in your tool, eg:
   - in Book Store Navigator app...
   - “Find science fiction books”

3a-b. Pick a Subgoal for that scenario. eg:
   Subgoal #1: “See bookstore map”.
   Q: Will Abby have formed this sub-goal?...
   • Yes/no/maybe.
   • Why? Consider Abby’s Motivations...

3c-d. Pick an Action for that subgoal.
   Action #1: “Tap ‘Browse Off’”:
   Q1. Will Abby know what to do?
   • Yes/no/maybe.
   • Why? Consider Abby’s... Tinkering

3e. Q2. If she performs the action, producing
   will Abby see progress toward the subgoal?
   • Yes/no/maybe. Why? Consider Abby’s Self-Efficacy &...
User Centered Design

User-centered design tries to optimize the product around how users can, want, or need to use the product, rather than forcing the users to change their behavior to accommodate the product.

-Wikipedia
Agile

User Centered Design

Agile customer representative

Manifesto for Agile Software Development

We are uncovering better ways of developing software by doing it and helping others do it. Through this work we have come to value:

Individuals and interactions over processes and tools

Customer collaboration over contract negotiation

Responding to change over following a plan

That is, while there is value in the items on the right, we value the items on the left more.

Kent Beck
Mike Beedle
Arie van Bennekum
Alistair Cockburn
Ward Cunningham
Martin Fowler

James Grenning
Jim Highsmith
Andrew Hunt
Ron Jeffries
Jon Kern
Brian Marick

Robert C. Martin
Steve Mellor
Ken Schwaber
Jeff Sutherland
Dave Thomas
2. Does my software amplify positive or negative behavior for users and society at large?
What if...
https://pair-code.github.io/what-if-tool/

What If...
you could inspect a machine learning model, with minimal coding required?
What if...

https://pair-code.github.io/what-if-tool/
Explain “why” to customers
What Instagram removing likes may mean for influencers and our self-esteem

SCIENCE & TECH - FEATURE

The decision could have a positive impact on the way people use the platform, but harm those trying to use it professionally.
Anıl Dash on how to prevent abuse


You should have real humans dedicated to monitoring and responding to your community.

You should have community policies about what is and isn’t acceptable behavior.

Your site should have accountable identities.

You should have the technology to easily identify and stop bad behaviors.

You should make a budget that supports having a good community, or you should find another line of work.
Deon
https://github.com/drivendataorg/deon

An ethics checklist for data scientists

deon is a command line tool that allows you to easily add an ethics checklist to your data science projects. We support creating a new, standalone checklist file or appending a checklist to an existing analysis in many common formats.

δέον • (déon) [n.] (Ancient Greek) [wiktionary]

Duty; that which is binding, needful, right, proper.
AI Incident Database

Welcome to the AIID

Why "AI Incidents"?

Intelligent systems are currently prone to unforeseen and often dangerous failures when they are deployed to the real world. Much like the transportation sector before it (e.g., FAA and FARS) and more recently computer systems, intelligent systems require a repository of problems experienced in the real world so that future researchers and developers may mitigate or avoid repeated bad outcomes.

What is an Incident?

The initial set of more than 1,000 incident reports have been intentionally broad in nature. Current examples include,
3. Will my software’s quality impact the humanity of others?
Quality has long been considered

Quality attributes

Notable quality attributes include:

- accessibility
- accountability
- accuracy
- adaptability
- administrability
- affordability
- agility (TfI) (see Common Subsets below)
- availability
- autonomy (Br)
- efficiency
- efficiency
- modifiability
- observability
- operability
- orthogonality
- portability
- precision
- predictability
- process capabilities
- producibility
- provability
- recoverability
- relevance
- reliability
- repeatability
- reusability
- resilience
- responsiveness
- resiliency (Br)
- robustness
- safety
- scalability
- self-sustainability
- serviceability (a.k.a. supportability)
- security
- simplicity
- stability
- standards compliance
- survivability
- sustainability
- tolerance
- testability
- timeliness
- traceability
- transparency
- ubiquity
- understandability
- upgradability
- vulnerability
- usability
Engineering ethics.

Ethics applies and is formalized in many professional fields: medical, legal, business, and engineering.

The first codes of engineering ethics were formally adopted by American engineering societies in 1912-1914. In 1946 the National Society of Professional Engineers (NSPE) adopted their first formal Canons of Ethics.

“hold paramount safety, health and welfare of the public”

Citigroup Center, Designed by Structural engineer William LeMessurier

Followed calculations required by building codes

Civil Engineering student Diane Hartley realized there was a problem

Tests showed that winds needed to bring it down would happen every 55 years
Professional Ethics

Professional ethics encompass the personal, and corporate standards of behavior expected by professionals.

First three “professions”

- Divinity
- Law
- Medicine
Hippocratic Oath ~450BC
“Do no Harm”
Law - Extrinsic

Bar regulates behavior

Oath to follow rules

Malpractice
Legal Malpractice

Not every mistake is legal malpractice. For malpractice to exist:

Attorney must handle a case inappropriately

due to negligence or with intent to harm

And cause damages to a client
Malpractice vs. Negligence

Negligence is a failure to exercise the care that a reasonably prudent person would exercise in like circumstances.

Malpractice is a type of negligence; it is often called "professional negligence". It occurs when a licensed professional (like a doctor, lawyer or accountant) fails to provide services as per the standards set by the governing body ("standard of care"), subsequently causing harm to the plaintiff.
Bioengineering Ethics:

- Respect for Autonomy
- Beneficence
- Nonmaleficence
- Justice
Professional Engineers

What {is / could be} the role of professional engineers in software?

https://en.wikipedia.org/wiki/Engineer%27s_Ring

By ----PCstuff 03:47, 31 July 2006 (UTC), CC BY-SA 2.5,
https://commons.wikimedia.org/w/index.php?curid=10340855
Will software quality impact human flourishing?

Most traditional emphasis of “engineering ethics”
What can we learn from other professions?
Should software have ”Professional Engineers“?
How do we define “safety critical systems”?
How much testing is enough? How can we convince others to do that much testing?
These questions are the start of the conversation, but as technology evolves, we must be vigilant to ensure we are promoting human flourishing.
Three questions to promote human flourishing

1. Does my software respect the **humanity** of the **users**?

2. Does my software **amplify positive** behavior, or **negative** behavior for users and society at large?

3. Will my software’s **quality** impact the **humanity** of others?
https://www.ted.com/talks/kevin_slavin_how_algorithms_shape_our_world?language=en
Key Points

• Awareness of ethical issues in software engineering
• Reflection on decision making
• Questions to ask when evaluating the ethics of software
• Starting points to dig deeper