Performance Reviews, Design Reviews, and Code Reviews
Project Peer Reviews

- Graded as a homework
  - Need to paint a picture of each member’s contribution
  - Need to be honest
  - Will get very low grade if don’t put in effort to differentiate contributions

- We know different people contributed in different ways at different times

- We know contributions are uneven and, within reason, as long as you felt people were good to excellent team members, we’re okay.

- Used qualitatively. Won’t be averaged, etc.

- Most people will get same score as team. Only poor team members will receive a penalty.

- We don’t want to make enemies of friends. Competition within teams is not healthy or productive.
Common practice for a long time was for reviews to be written by a “Manager”

- Perception that manager was expert, unbiased, had full-visibility, and perspective
- Clear single point of failure
  - Conscious bias
  - Unconscious bias
- No way to escape unfairness, except leave team
- Managers don’t always know internals of how team supports each other
- On the bright side, reviews stimulated conversation, which can lead to growth
More recently, peer reviews began informing or replacing manager-only reviews
- More perspectives = More information
- More diversity = More fairness
- Often times peers and mentors/seniors peers

When used in a formulaic way for promotion, merit raises, etc, can lead to competitiveness and backstabbing
- Loss of teamwork
- Loss of feeling of safety within team
- Unwillingness to take risks
- Loss of creative engine among typical employees

Still much concern bias, especially unconscious bias
- Training programs improve hiring and team dynamics as well as quality of peer reviews
Better Way to Develop Talent

- Peer reviews aren’t used formulaically, except for institutional purposes.
- Peer reviews help managers coach talent and assign best opportunities for contribution and growth.
- Good to excellent reviews are not at risk. Poor performers are, obviously, a concern.
- Promotion not based on highest scores, etc.
Better Model for Promotion

- All Good to Excellent contributors are valued

- Promotion is a recognition that one is already doing the higher-level job.
  - One can’t ask for promotion.

- “Stretch Opportunities” lead to growth and promotion.
  - If manager feels you are ready, sometimes after prompting from you, the manager will try to find appropriate opportunities for you.
    - Success leads to more opportunities and promotion
  - If you ask and are not ready, managers will explain the types of growth you need and try to find opportunities where you can be mentored to achieve it
    - The mentors on these opportunities provide a second data point about readiness for “stretch opportunities” that can lead to promotion
Design Review: Goals

- Make sure design meets all requirements
- Make sure design is high quality
  - Best practices, Maintainability, Symmetry with other projects, Etc
- Make sure design is documented appropriately
- Make sure all of those who will be impacted by design have a chance to be heard
Design Review: How To

- **Common technique**
  - Design documents posted for review and moderator/shepherd assigned
  - Moderator picks diverse reviewers
    - Experienced and respected software engineers
    - Those who know domain best (may not even be coders in some cases, e.g. mechanical or electrical engineer who knows hardware best)
  - Reviewers read and post concerns
  - Software Engineer reviews and responds to concerns
    - Adjusts design or documentation or explains why not
  - Physical or virtual review meeting to review disposition of all changes
    - Sometimes need to mediate conflicting advice
  - Design is accepted
Code Review: Goals

- Verify code quality, not just correctness
  - Conformance to code standards, quality of documentation, other elements of maintainability

- Finds some functional defects missed by testing
  - Finds cases where code and test set are both wrong

- Encourages good practices, because everyone knows they are “being watched”

- Domain specialists can check really nuanced logic, etc.

- Opportunity for new developers to see best practices, how ecosystem is used, etc.

- Opportunity for seasoned developers and developers absorbed through M&A to be reminded of standards by recently on-boarded engineers.

- Growth through learning-by-example
Common technique

- Code posted for review and moderator/shepherd assigned
- Moderator picks diverse reviewers
  - Those who know domain best (may not even be coders in some cases, e.g. mechanical or electrical engineer who knows hardware best)
  - Seasoned and less experienced
  - Sometimes just “who is available”
- Reviewers read and post concerns
- Developer reviews and responds to concerns
  - Adjusts code or explains why not
- Physical or virtual review meeting to review disposition of all changes
  - Sometimes need to mediate conflicting advice
- Code is accepted
Coding standards

- Document and standardize best practices
- Ensure consistency, which leads to better readability and maintainability
- Provide a standard for code reviews
- Influenced by culture, ecosystem, etc.

Examples:
- [https://google.github.io/styleguide/javaguide.html](https://google.github.io/styleguide/javaguide.html)
- [https://source.android.com/source/code-style.html](https://source.android.com/source/code-style.html)
Design and Code Review: Agile Disciplines

- Usually no formal meetings or documentation
- Usually based on tight collaboration among team

Example: Extreme Programming

- “Driver” and “Navigator” talk through the plan
  - Plan design
  - Identify special cases and risky elements
    - Walk through these carefully
- “Driver codes” while “Navigator” reviews code
- “No professional drivers”
  - “Driver” and “Navigator” take turns to maintain vigilance and involvement.
- Provides for design review, code review, and feeling of “someone watching, so it matters”