Module 4

Human Aspects of Software Engineering

▼ What traits make up a software engineer?

- Individual responsibility: The drive to deliver on promises to peers, stakeholders, and management.
- Awareness: Understanding the needs of other team members, of stakeholders, and of the managers who oversee the project.
- Brutal honesty: Pointing out flaws in an idea or a program design in an honest and constructive manner.
- Resilience under pressure: Handling the pressure of changing requirements and priorities as well as demands from stakeholders and managers.
- Sense of fairness: Sharing credit with coworkers, avoiding conflicts of interests, and valuing the work of others.
- Attention to detail: Detailed attention to the criteria established for the project (e.g., performance, cost, quality).
- Pragmatism: Recognizing that while software engineering is a discipline with rules, these rules can be adapted to existing circumstances.

▼ What are some constructive feedback tips?

- **Balance the Positive and the Negative**
  - When giving constructive criticism, it's important to make sure you're presenting a balanced perspective, whether your feedback is ultimately positive or negative. This is more obvious when it comes to negative feedback - while you shouldn't have to feel like you must paint a picture

that's different from the reality of the situation, especially if you have major concerns about the work or behaviors being discussed, it's helpful to be able to point out some positives in that person's attitude or output. For example, if a specific project doesn't meet your expectations, you could frame the conversation by saying how you've been impressed with the individual's work in the past, which is why you know that this deliverable could be improved. Again, you want to be truthful - don't mislead someone into thinking their performance is better than it actually is - but giving someone a few positives to help motivate them can go a long way.

- Be problem-focused and specific. An important part of telling an employee what they could do better is to tell them why.
- Talk about the situation, not the individual.
- Give praise where it's due.
- Be direct but informal.
- Be sincere.
- Listen.
- Make it timely.

▼ How can a team establish a **team philosophy** that will bind them together and create a sense of belonging?

through a team charter

▼ Why is trust necessary on a team?

- Trust inspires **confidence.**
- When team members are confident that they can count on each other, work goes well.
- To be specific, software engineers should trust the skills and competence of their teammates and their manager.
- When they do so, they typically feel comfortable asking for assistance when they need it.
- An environment of trust and confidence in themselves and each other makes a team successful and will affect overall software quality.

▼ How can teams hold team members accountable?

- It is recommended that each team itself should establish its own performance management techniques for **accountability**.
- This can be done through technical reviews and memos.
- In addition, corrective action needs to be taken when a team member does not perform adequately.
- However, as is done in most workplaces, corrective action steps can be avoided by providing team members with timely and specific feedback.

▼ How is a team commonly setup?

- The team may be set up as one large team or several smaller teams.
- Each team should have a technical lead and a project manager.

▼ What decision should be made at the outset by a team?

From the beginning a decision should be made on which development tools to use although this decision may need to be modified as needed.

▼ What is the recommended approach to development?

It is currently recommended that the project team adopt an agile scrum approach to development, with the features clearly defined by product managers and organized for development by the project manager into stories.

▼ What factors should be considered when defining a team structure?

- The nature and difficulty of the problem
- The size of the program in lines of code or function points
- Time that the team will work together
- Degree to which the program can be modularized
- Reliability of the system being built
- Delivery date
- Communication needs for the project

▼ The social processes around software development are highly dependent on engineers' abilities to do what?

- find and connect with individuals who share similar goals and complementary skills
- harmonize each team member's communication and teaming preferences
- collaborate and coordinate during the entire software lifecycle
- advocate for their product's success in the marketplace