# Teams and teamwork

- All SE methods and processes are about coordinating teamwork and teams
  - Agile : **people and interaction** over processes
  - Customer **collaboration** over contract negotiation

## Traditional view

- Have a requirements team

- Have a system design/architecture team

- Have a development team

- Have a testing team

- Have a release team

- Have client-relationship managers

## Choosing a project team

- Team chosen for the project vs Project chosen for the team
- You need the right balance of skills
- But not too big (more than 10 people makes it too hard)

## Agile teams

- Project./Tech leader ( Scrum master)

  - Facilitating the team, organising, motivating, leading

- Team members (mix of skills)

  - Designers, developers, database architects, testers

- Product owner (representing clients needs in team)

  - Manages prioritisation of requirements/ user stories/ backlog

- Stakeholder (active involved person from client/ users)

- Big companies may have several agile teams working on portions of a product

## Support Roles

- Companies often have resource-people

  - Build/configuration expert who sets up scripts for teams/projects

  - Domain experts (medical/legal requirements)

  - Hardware experts

  - Independent testers (black-box testers)

- Agile team pull on these resources as needed

- Also e.g. Quality assurance teams that sit above project teams

**Managers / Project leaders**

- Who should lead a project

- Managers often work with a project leader

  - Should recognise that the project leader knows more than them about some things

  - Project leaders should recognise that team members have better knowledge than them on some things

- Managers should trust team members equally, respect people for their knowledge/skills, include everyone as much as possible, and be honest about things – all of this works both ways.

**Job vs. Role**

- The are people with **job titles** and **roles**  in the team

- Project/Tech leader is running a project but is a software engineer or people-manager

**Project leader decisions**

- Who should make the critical technological decisions?
- How is relationship with the client going to be managed?
- Who's job will it be to keep things on time-track?
- Is anyone else (from resource-people) needed for it?
- What's the best way to have a cohesive team on the project?
- How formal should organisation be?

**Motivated Individuals**

- People like becoming good at something, and working on tasks they **can** do well at
  - Developing a specialisation and using it in projects, and likewise teaching other new things
  - Learning from others in unfamiliar tasks
  - Sharing other 'necessary' work
- People tend to have a style
  - task-oriented (smart solutions and good code)
  - self-oriented (personal development and achievements)
  - interaction-oriented (collaboration and user focus)
- Social needs
  - Creating collaborative work, as well as social time outside work (especially for remote work)
  - Regular face-to-face meetings (that don't waste time)
- Esteem needs
  - Being respected for what they are good at (if they are trying)
  - Recognitions of achievements (peer awards, not just bonuses, etc.)
- Self-realisation needs
  - Responsibility for their work, specialising in certain skills
  - Opportunity to develop new ones on projects