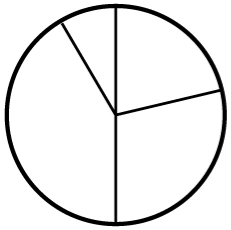
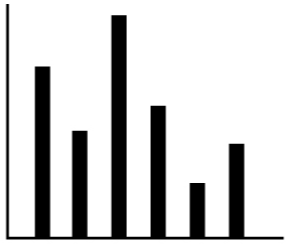
Monitoring

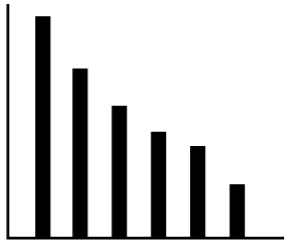
- Collect status information
- Analyze and evaluate information
- Present and communicate status



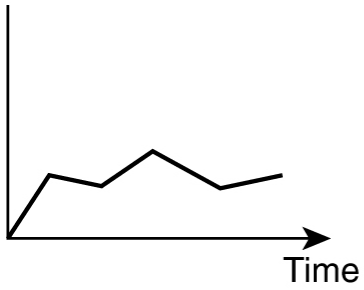
Pie chart



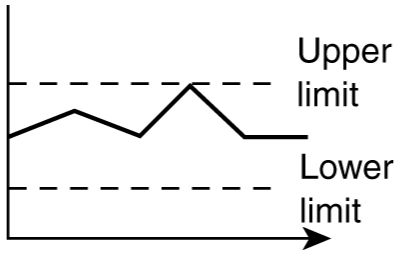
Histogram



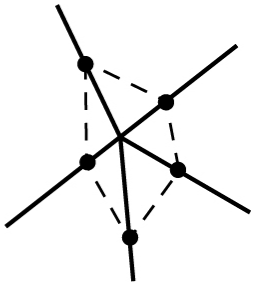
Pareto diagram



Time chart



Control chart

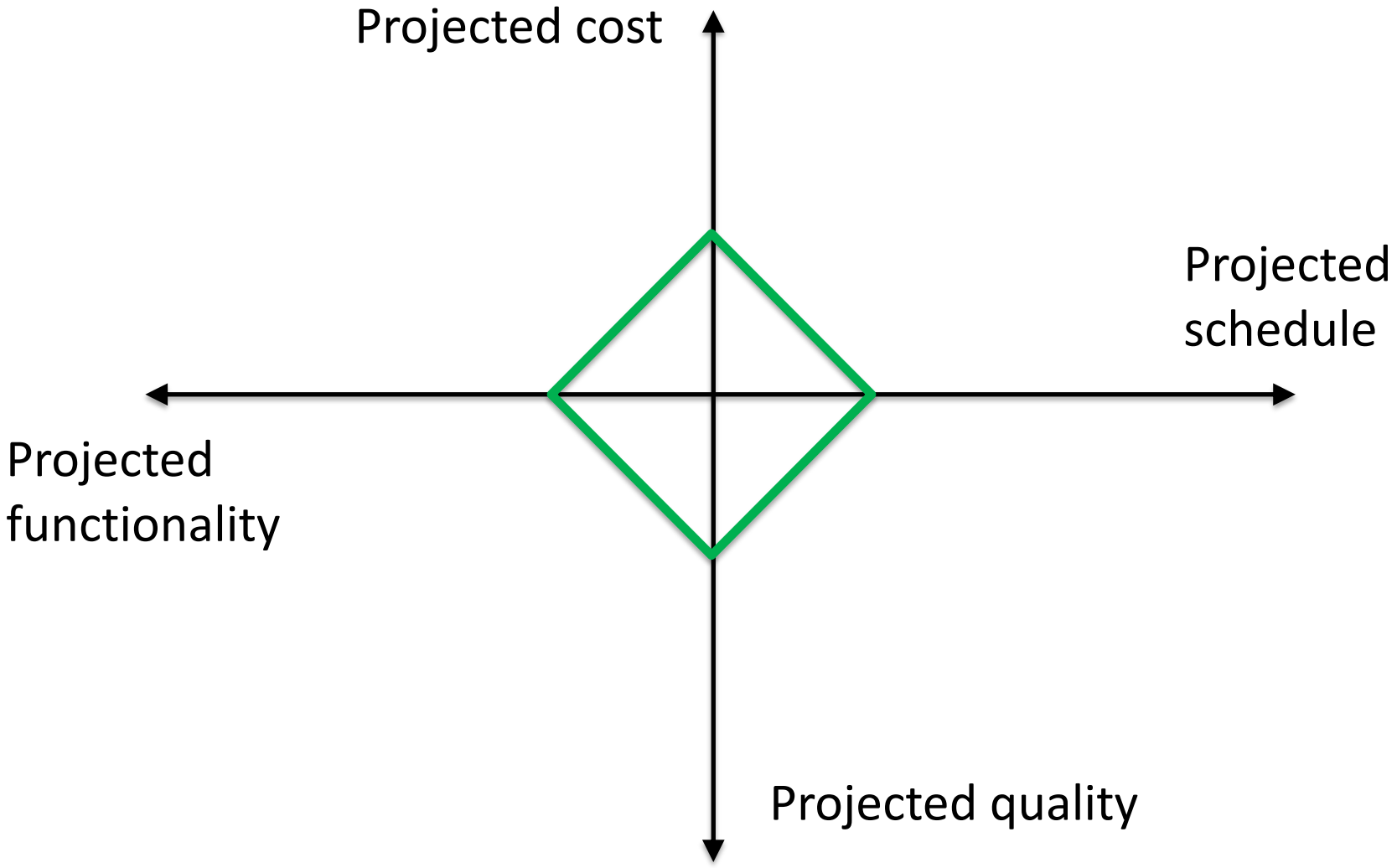


Kiviat chart

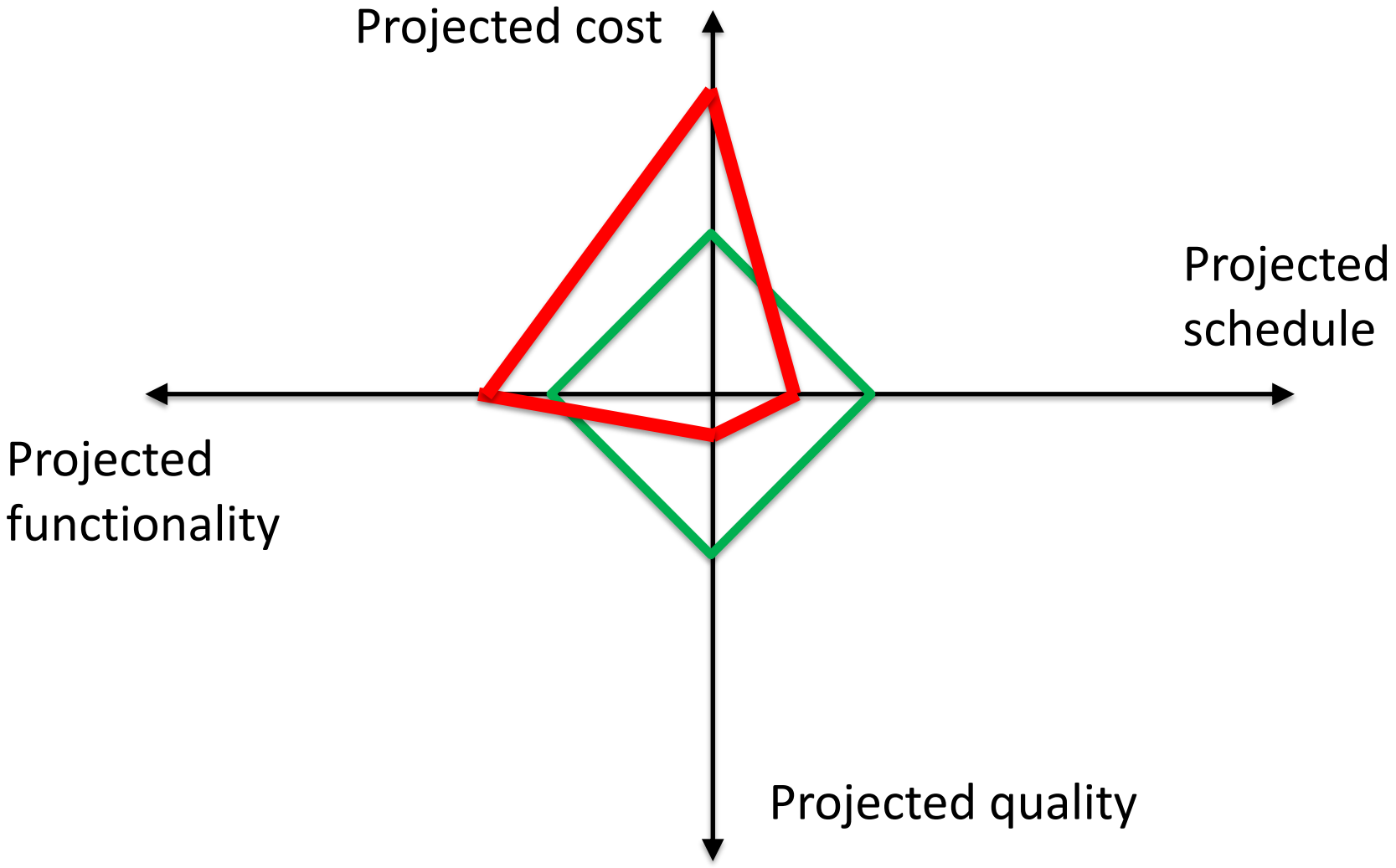
Adjusting

- What you can adjust
  - Resources (affects cost)
    - Bring in hired gun testers to run defined scripts
  - Schedule
    - Sometimes external conditions can push up project schedule
  - Project content
    - Delete functions to meet schedule
  - Quality

Adjusting (cont.)



Adjusting (cont.)



Effort Estimation Approaches

- Quick for OO implementations
  - Estimate number of classes to solve the problem
  - Categorize user interfaces
    - None 2.0
    - Simple text 2.25
    - GUI 2.5
    - Complex GUI 3.0
  - Multiply number of classes by weight associated with interface type
  - Multiply that by estimated person-days to develop a class (typically 15 – 20)

Effort Estimation Approaches (cont.)

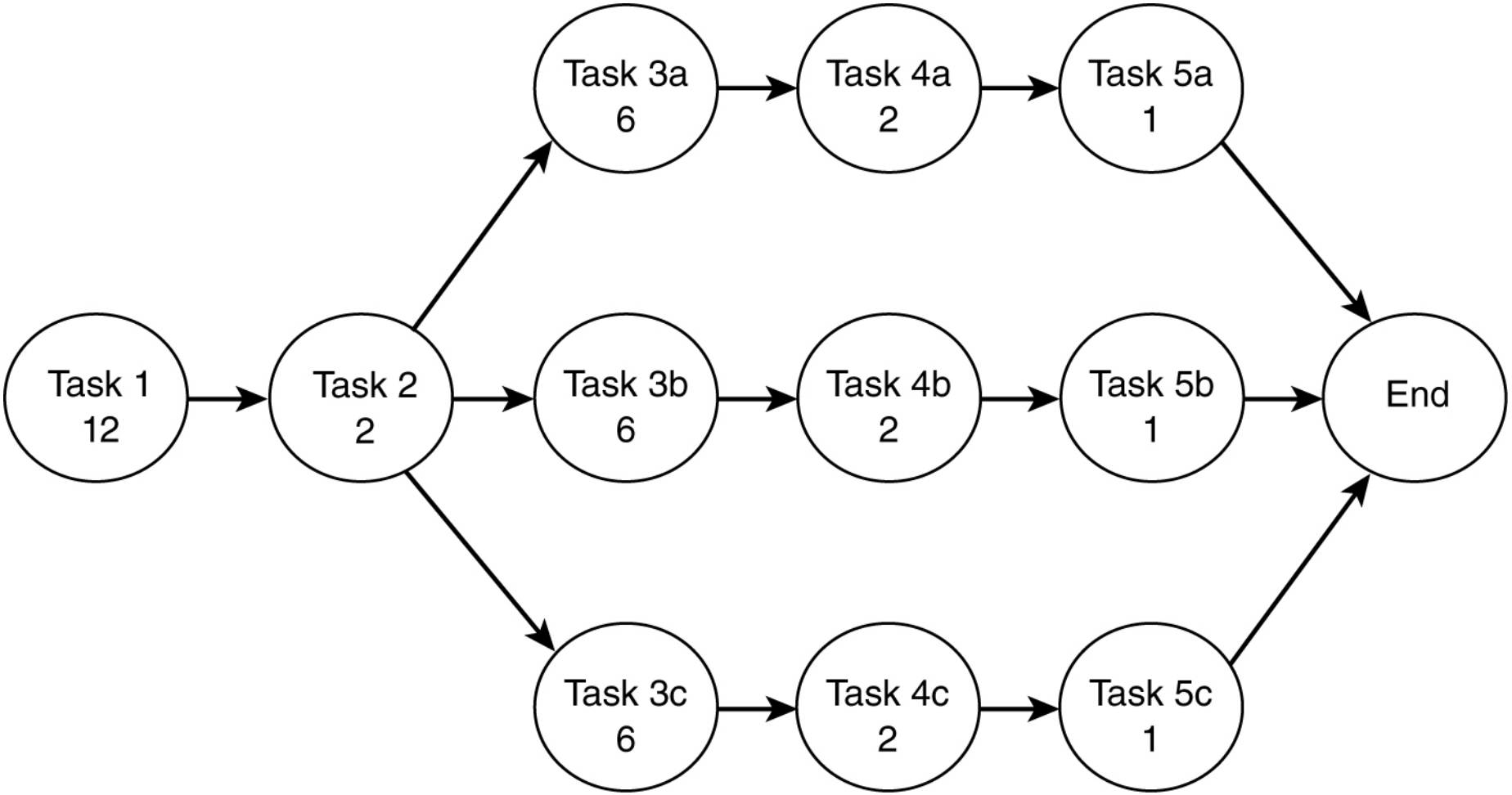
- Work Breakdown Structure
  - Determine required deliverables for project
  - Identify steps and tasks needed to implement each deliverable
  - Sequence tasks (identify parallelism)
  - Estimate effort to implement each task
  - Tentatively assign people to each task
  - Estimate time to complete each task
  - Create timeline showing resources and timing of tasks

WBS Example

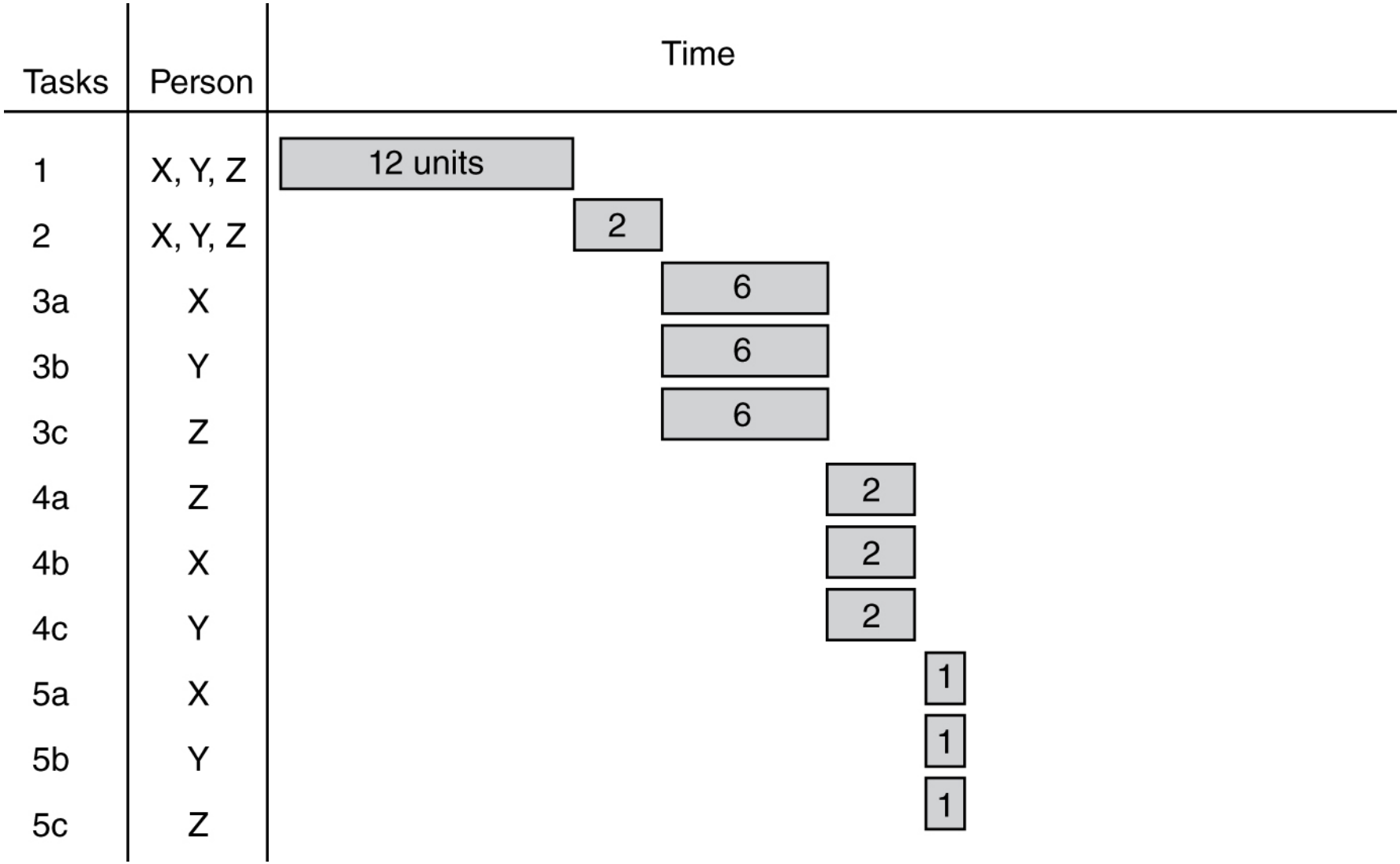
- Work is to develop test scenarios for project
- Suppose 5 tasks
  - Read and understand requirements
  - Develop list of test scenarios
  - Write the script for each scenario
  - Review each scenario
  - Modify and change the scenarios

Can be done in parallel

WBS Example (cont.)

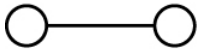


WBS Example (cont.)



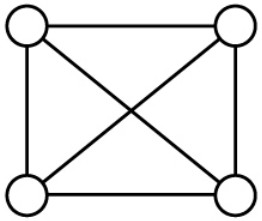
Team Management

- Systems typically developed by teams
- Team structure and dynamics are critical success factors



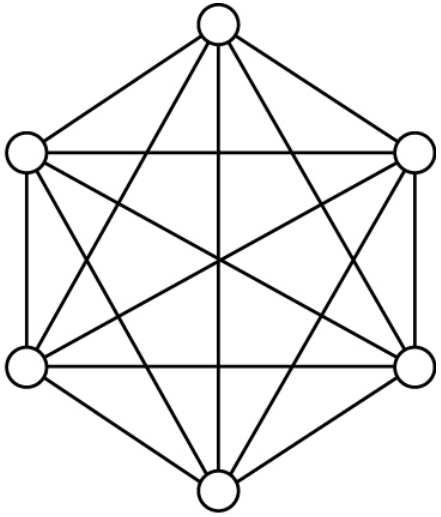
2 people:

1 path



4 people:

possibly 6 paths



6 people:

increase to potentially 15 paths

Team Size

- Lone developer
  - Isolation
  - No one to bounce ideas off
- Whole development organization is team
  - Too much communication, no time for actual writing
- Experience suggests **3 – 8 people** works for typical projects
  - Routine projects with well-defined responsibilities can go higher

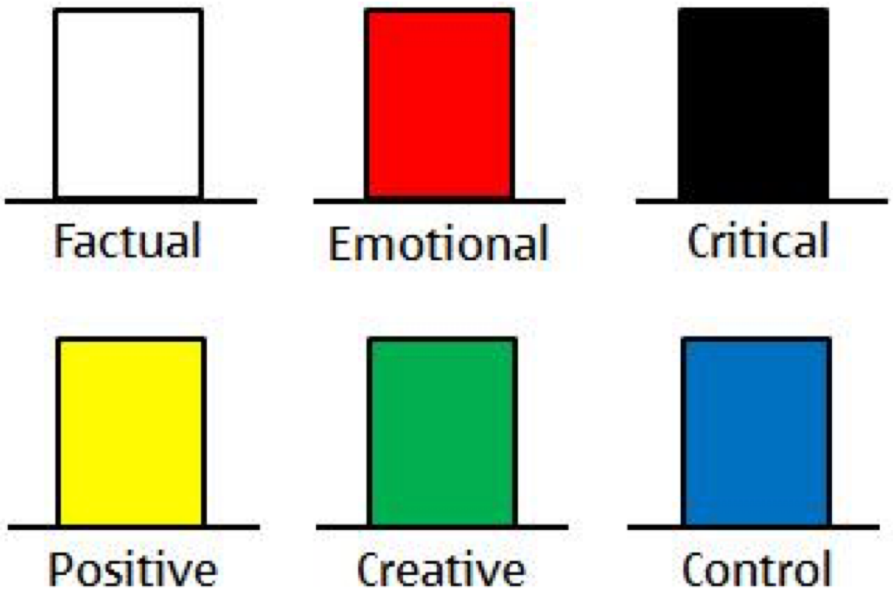
Inter-team Coordination

- Democratic decentralized
- Controlled centralized
- Controlled decentralized

Democratic Decentralized

- No real hierarchy – horizontal communication
- Everyone is a peer
- Whole group discusses issues
- Decisions reached by consensus
- **Disadvantage**: no one tasked to drive progress, potential time wasting
- **Advantage**: good at generating novel solutions
- Need some ground rules to keep team progressing, limit time for discussion

Six Thinking Hats



- Factual hat – focus on available data
- Emotional hat – depend on emotion and intuition
- Critical hat – find bad points
- Positive hat – find advantages
- Creative hat – brainstorming
- Control hat – force hat changes, summaries

6 Thinking Hats (cont.)

- Then use hats in different orders to accomplish goals
  - Initial ideas: blue, white, green, blue
  - Strategic planning: blue, yellow, black, white, blue, green, blue
  - Problem solving: blue, white, green, red, yellow, black, green, blue