5. Project planning and management

- Role of a manager
- Charts and Critical Path Analysis
- Estimation Techniques
- Monitoring

Role of a manager

- Directs resources for the achievement of goals
- LEADER also provides
  - Vision
  - Inspiration
  - Rises above the usual
- No one right way to manage

Management Continuum

Authoritarian **Democratic**

Autocratic **Consultative** **Participate**

Solves problems alone
Dictates decisions

Discusses Problems
Makes decision

Chairperson
Agrees problem
Creates consensus

Managerial Roles

» (after Henry Mintzberg)

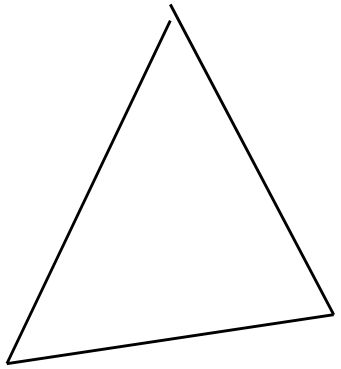
- Interpersonal
  - Figurehead
  - Leader
  - Liaison
- Informational Roles
  - Monitor
  - Disseminator
  - Spokesperson
- Decisional Roles
  - Entrepreneur
  - Resource Allocator
  - Disturbance Allocator
  - Negotiator

Qualities

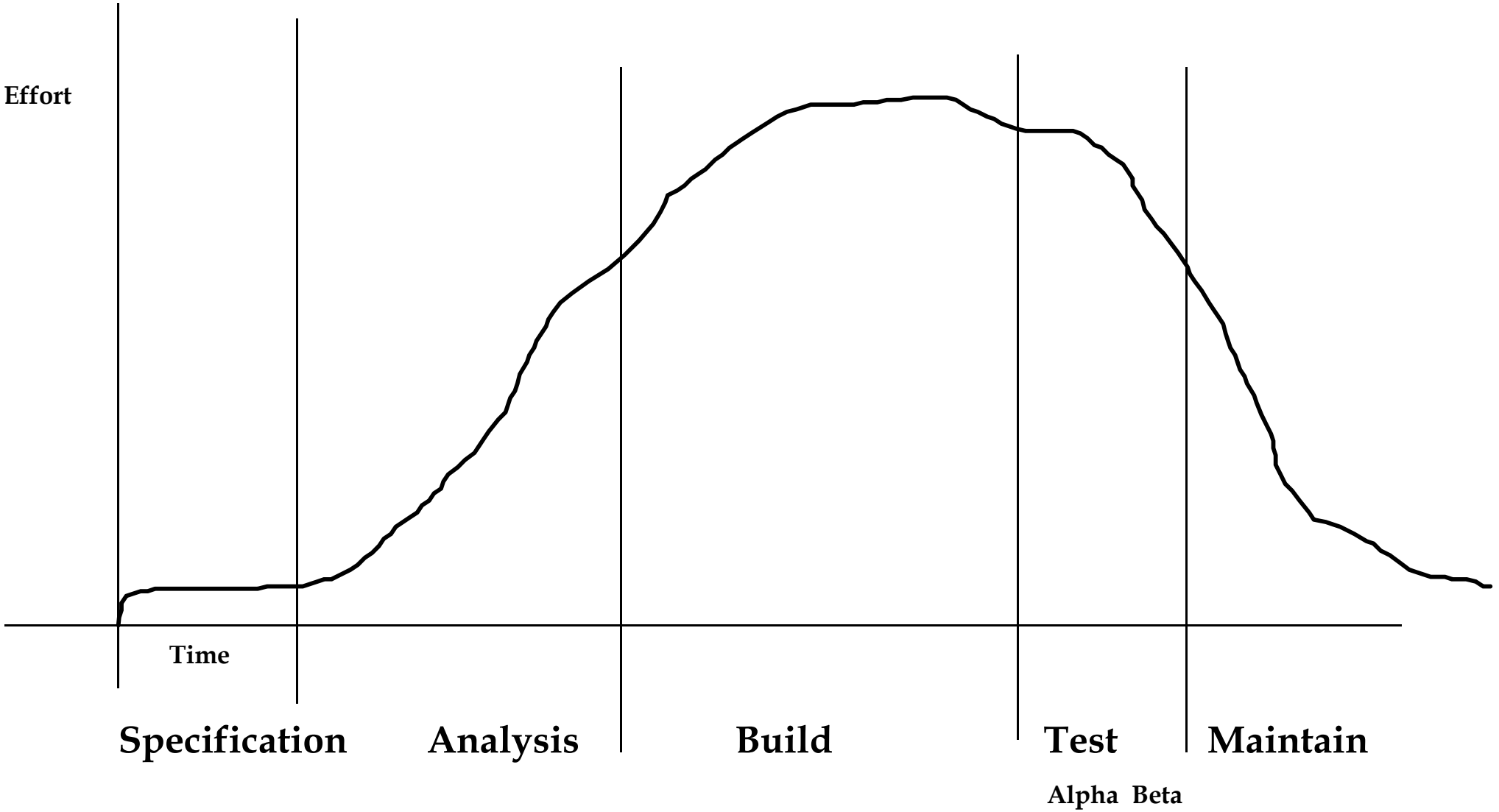
- Technical/Professional knowledge
- Organisational know-how
- Ability to grasp situation
- Ability to make decisions
- Ability to manage change
- Creative
- Mental flexibility - Learns from experience
- Pro-active
- Moral courage
- Resilience
- Social skills
- Self Knowledge

Variables

- Resource
- Time
- Function
- “ You can have any two of quick, good or cheap, but not all three”

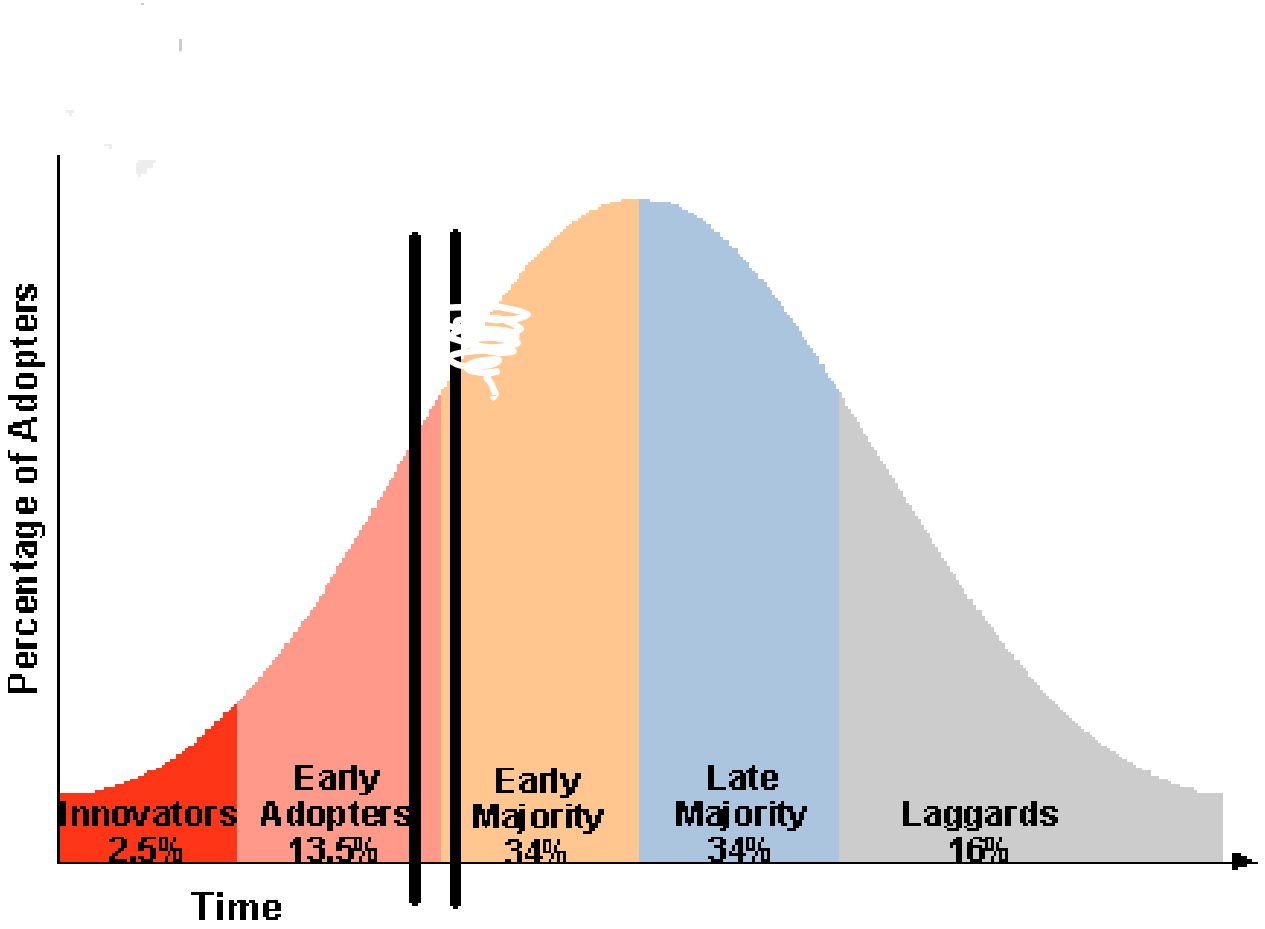


Development cycle:



Crossing the Chasm

- Geoffrey Moore, after Everett Rogers



Approaches and methodologies

- Top Down
  - Waterfall decomposition
- Bottom Up
  - meta machine
- Rapid Prototype
  - successive refinement
  - Agile Engineering
- Muddle through

Agile Engineering

In February 2001, 17 software developers[5] met at the Snowbird, Utah resort, to discuss lightweight development methods. They published the *Manifesto for Agile Software Development*[1] to define the approach now known as agile software development. Some of the manifesto's authors formed the Agile Alliance, a nonprofit organization that promotes software development according to the manifesto's principles.

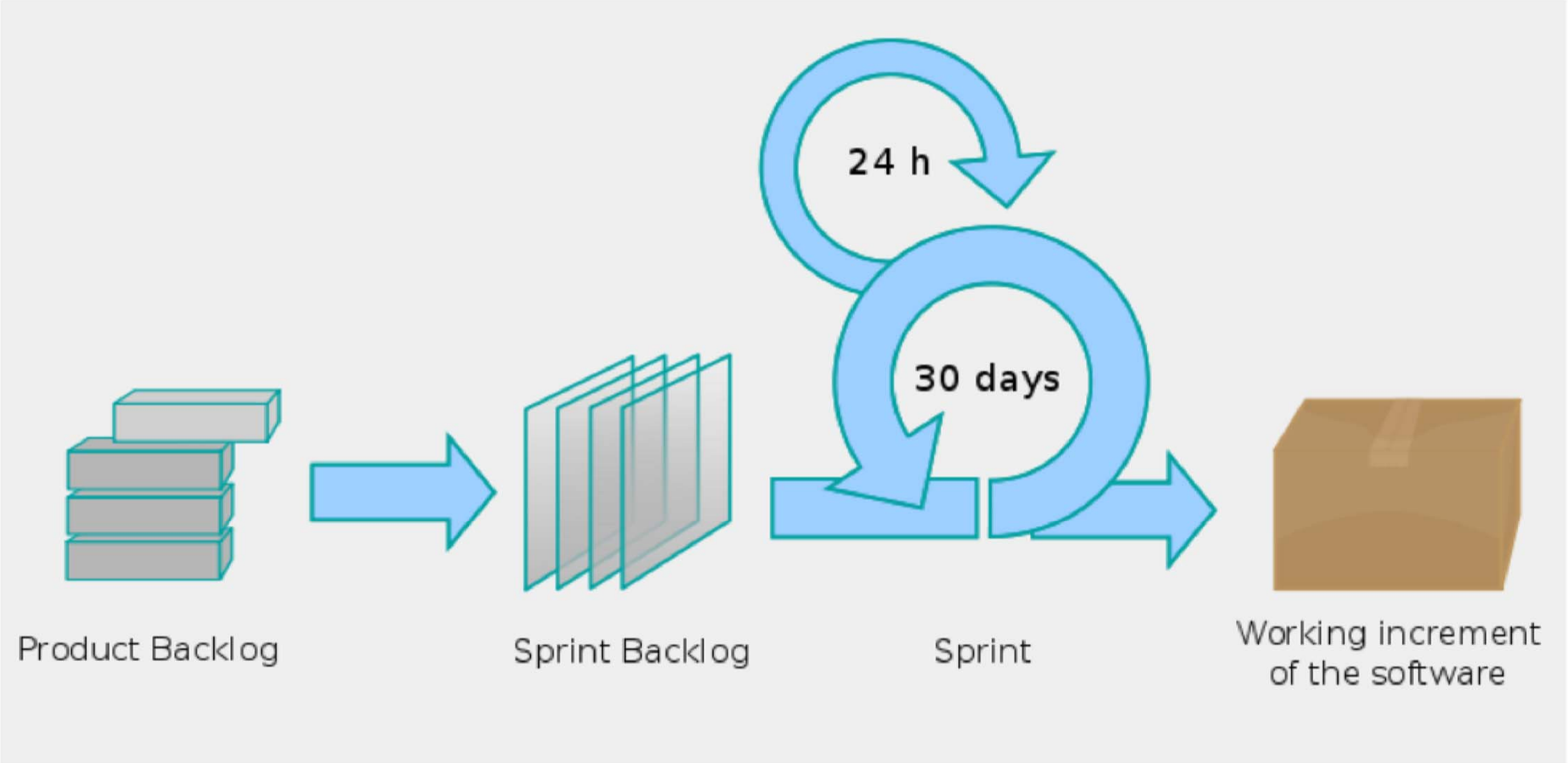
The Agile Manifesto reads, in its entirety, as follows:[1]

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
- Working software over comprehensive documentation
- Customer collaboration over contract negotiation
- Responding to change over following a plan

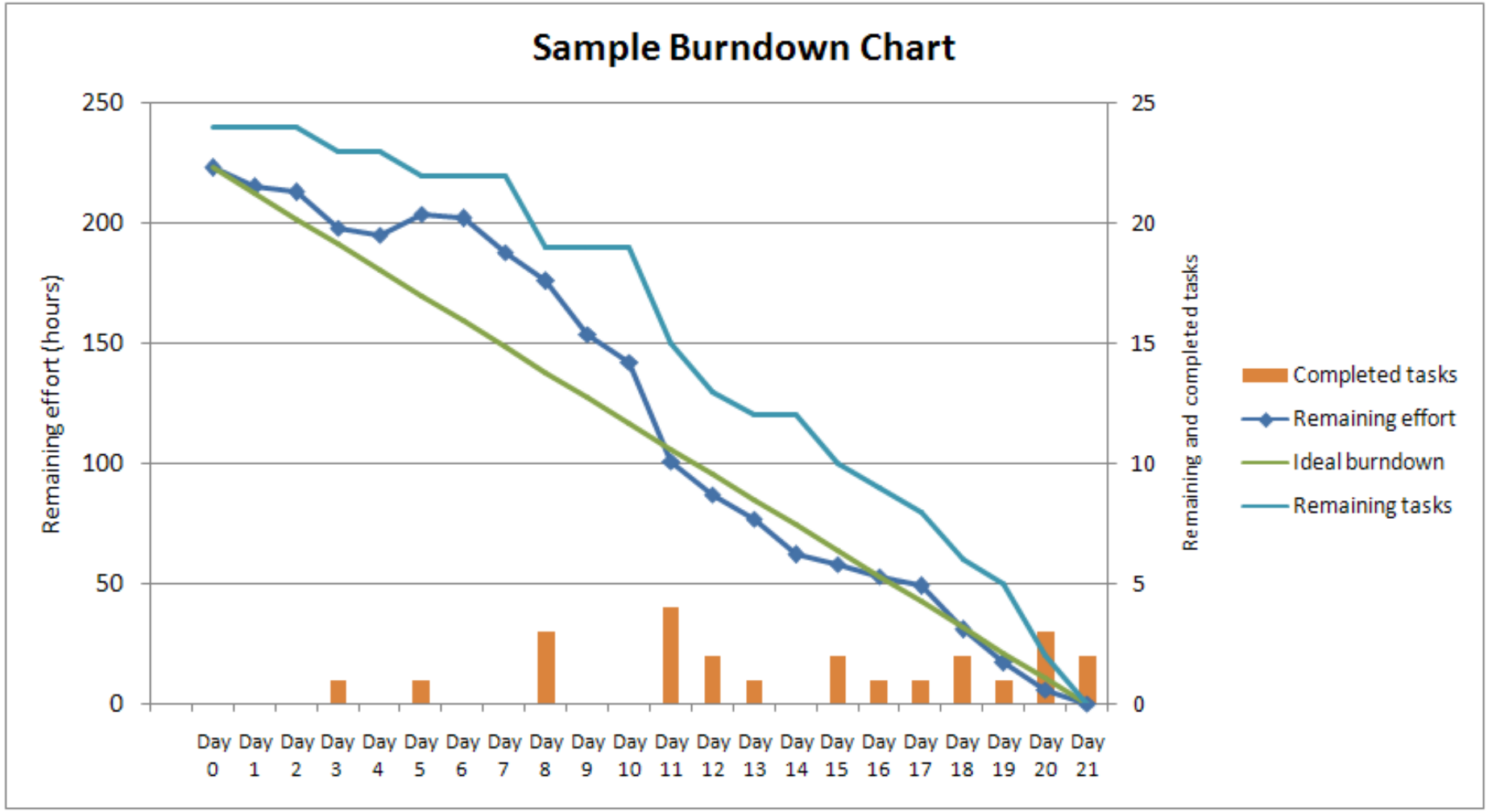
That is, while there is value in the items on the right, we value the items on the left more

Sprints, Scrums, Timeboxes



DeGrace, Peter; Stahl, Leslie Hulet (1990-10-01). *Wicked problems, righteous solutions*. Prentice Hall. ISBN 978-0-135-90126-7.

Adapted from http://en.wikipedia.org/wiki/Scrum_(development)


Sample Burndown Chart
Remaining effort (hours)
250
200
150
100
50
0
25
20
15
10
5
0
Remaining and completed tasks
Day 0
Day 1
Day 2
Day 3
Day 4
Day 5
Day 6
Day 7
Day 8
Day 9
Day 10
Day 11
Day 12
Day 13
Day 14
Day 15
Day 16
Day 17
Day 18
Day 19
Day 20
Day 21
Completed tasks
Remaining effort
Ideal burndown
Remaining tasks

Spiral Methodology

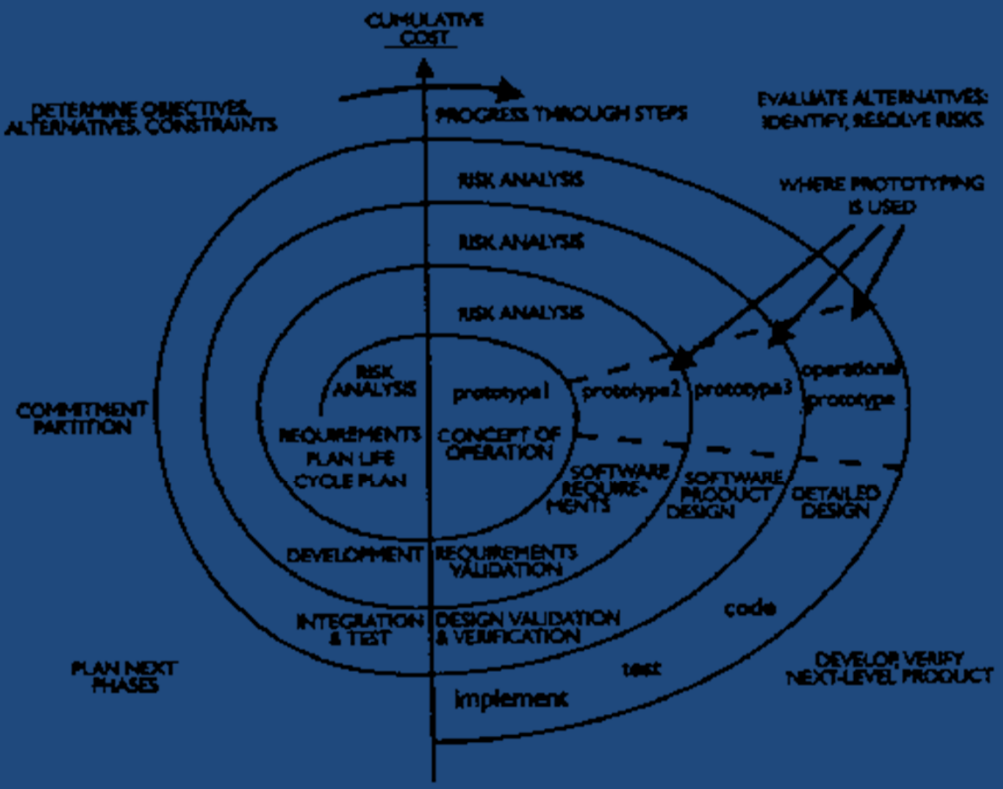
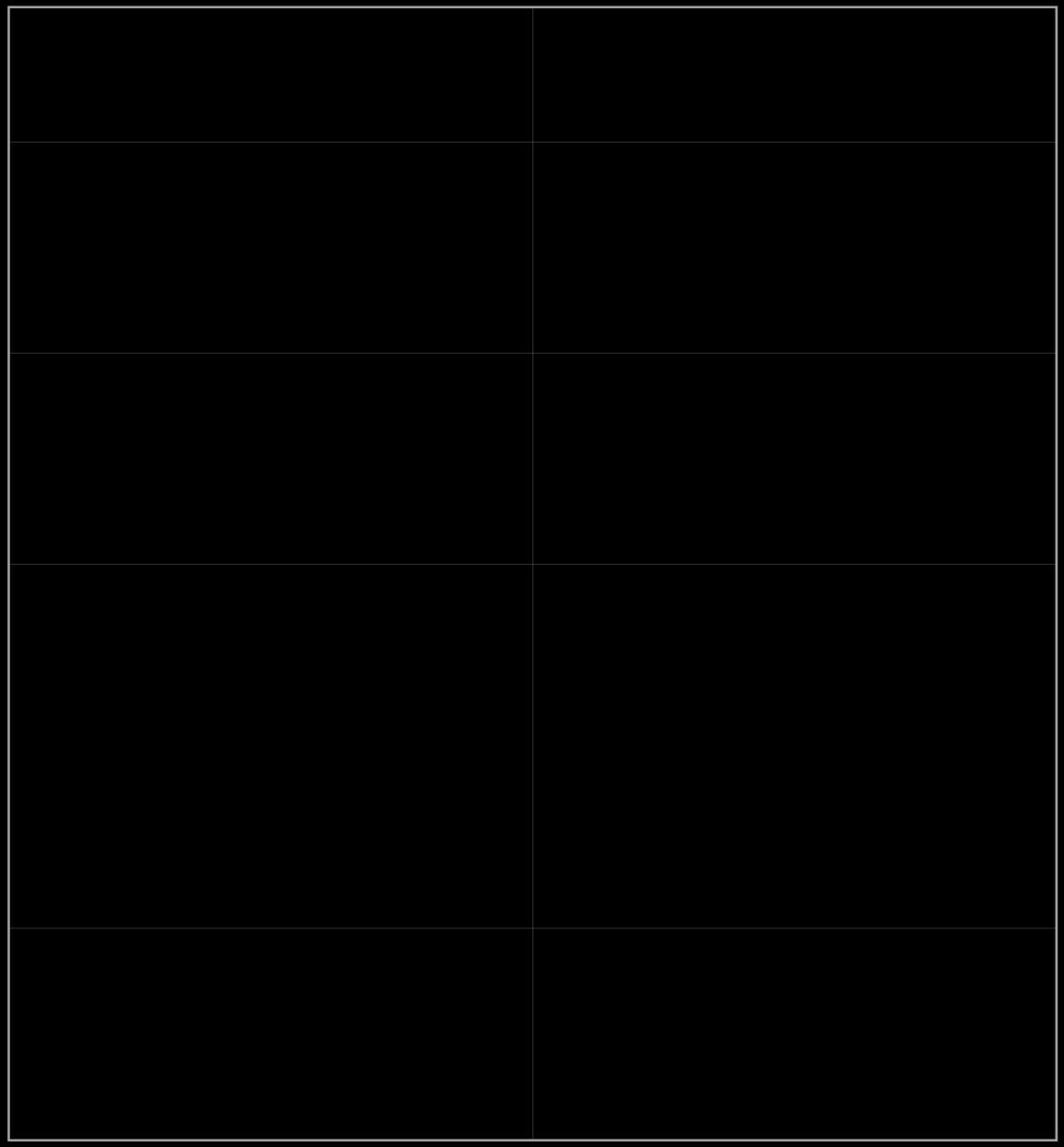


FIGURE 4: BOEHM SPIRAL MODEL (REPRINTED WITH PERMISSION FROM [10]).



Meetings

- Daily Scrum
  - Each day during the sprint, a project status meeting occurs. This is called a *daily scrum*, or *the daily standup*. This meeting has specific guidelines:The meeting starts precisely on time. All are welcome, but normally only the core roles speak The meeting is timeboxed to 15 minutes
- Scrum of scrums
- Each day normally after the daily scrum.These meetings allow clusters of teams to discuss their work, focusing especially on areas of overlap and integration. A designated person from each team attends.

- Sprint Planning Meeting
  - At the beginning of the sprint cycle (every 7–30 days), a "Sprint Planning Meeting" is held.Select what work is to be done. Prepare the Sprint Backlog that details the time it will take to do that work, with the entire team. Identify and communicate how much of the work is likely to be done during the current sprint. Eight hour time limit
    - (1st four hours) Product Owner + Team: dialog for prioritizing the Product Backlog
    - (2nd four hours) Team only: hashing out a plan for the Sprint, resulting in the Sprint Backlog
- At the end of a sprint cycle:
- Sprint Review Meeting
  - Review the work that was completed and not completed. Present the completed work to the stakeholders (a.k.a. "the demo"). Incomplete work cannot be demonstrated. Four hour time limit
- [Sprint Retrospective
  - All team members reflect on the past sprint. Make continuous process improvements. Two main questions are asked in the sprint retrospective: What went well during the sprint? What could be improved in the next sprint? Three hour time limit

Pert and Gantt Charts

- Visual representation of project
- Microsoft Project

Example: Getting up in the morning

Task	Duration (mins)
1 Alarm rings	0
2. Wake Up	3
3. Get out of bed	5
4. Wash	5
5. Get dressed	5
6. Put kettle on	2
7 Wait for kettle to boil	5
8 Put toast on	2
9 Wait for Toast	3
10 Make coffee	3
11 Butter Toast	2
12 Eat Breakfast	10
13 Leave for Lectures	0

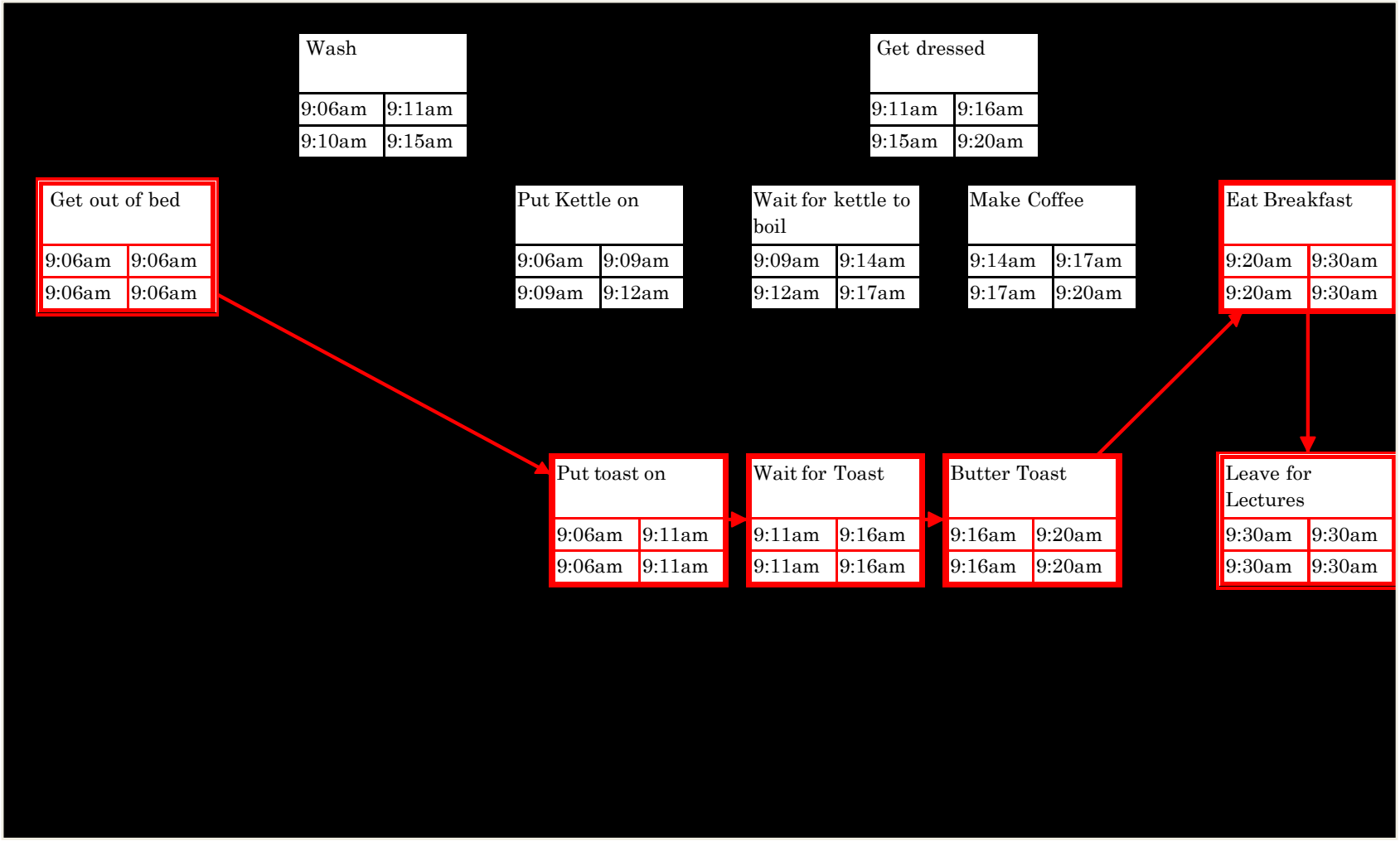
Pert Chart


Get out of bed
8:56am 8:56am
8:56am 8:56am
Wash
8:56am 9:01am
8:56am 9:01am
Get dressed
9:01am 9:06am
9:01am 9:06am
Put Kettle on
9:06am 9:09am
9:09am 9:12am
Wait for kettle to boil
9:09am 9:14am
9:12am 9:17am
Make Coffee
9:14am 9:17am
9:17am 9:20am
Eat Breakfast
9:20am 9:30am
9:20am 9:30am
Put toast on
9:06am 9:11am
9:06am 9:11am
Wait for Toast
9:11am 9:16am
9:11am 9:16am
Butter Toast
9:16am 9:20am
9:16am 9:20am
Leave for Lectures
9:30am 9:30am
9:30am 9:30am

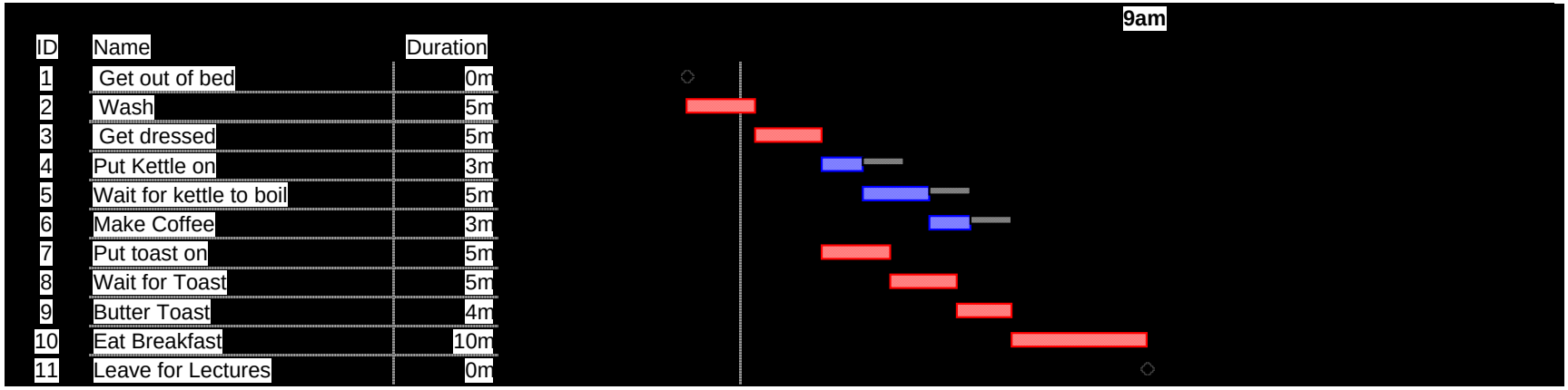
Critical Path Analysis

- Compute earliest and latest start/finish for each task
- The difference is the *slack*
- The Critical Path joins the tasks for which there is no slack
- Any delay in tasks on the on the critical path affects the whole project

Pert Chart

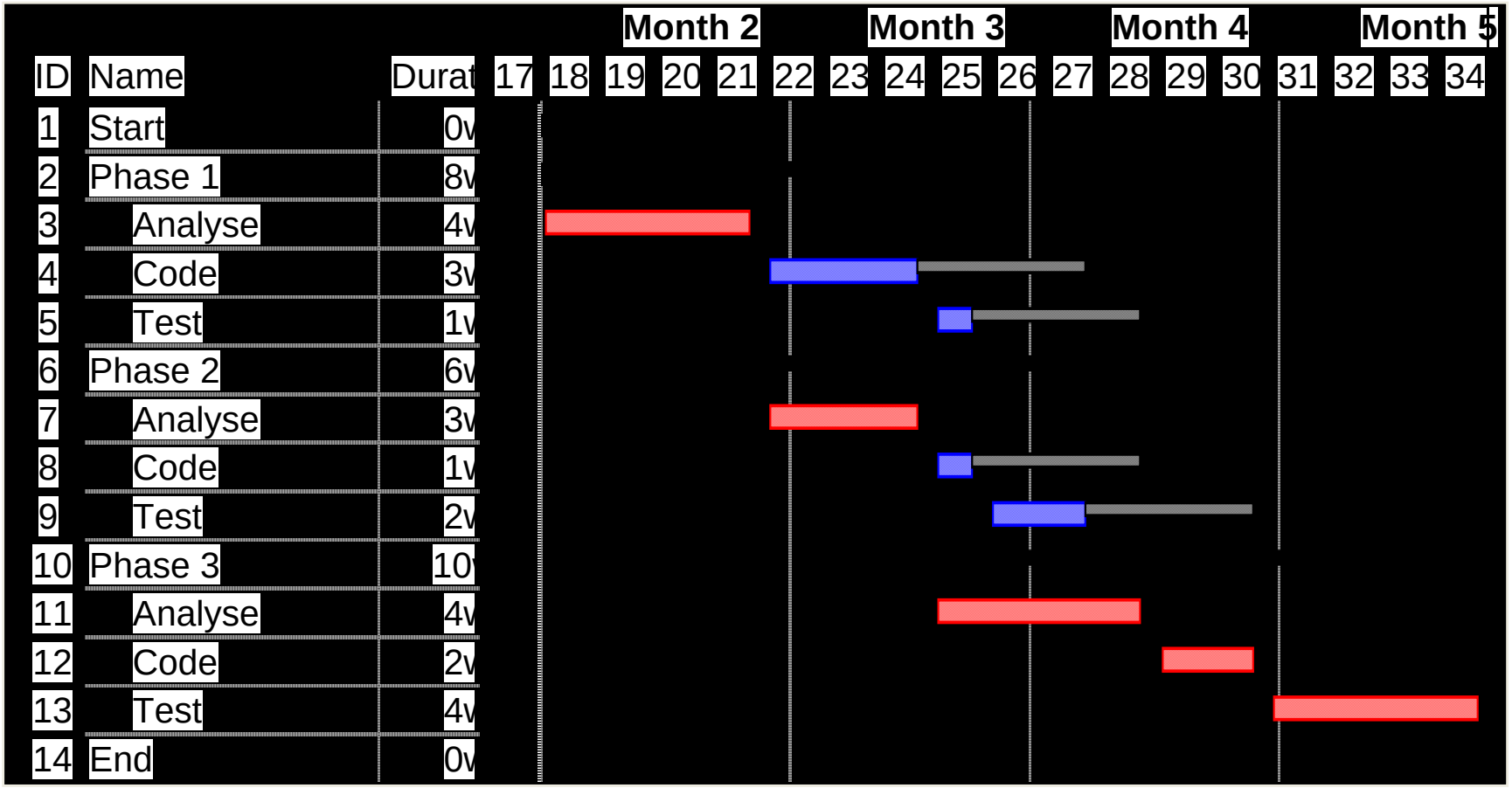

Wash
9:06am 9:11am
9:10am 9:15am
Get dressed
9:11am 9:16am
9:15am 9:20am
Get out of bed
9:06am 9:06am
9:06am 9:06am
Put Kettle on
9:06am 9:09am
9:09am 9:12am
Wait for kettle to boil
9:09am 9:14am
9:12am 9:17am
Make Coffee
9:14am 9:17am
9:17am 9:20am
Eat Breakfast
9:20am 9:30am
9:20am 9:30am
Put toast on
9:06am 9:11am
9:06am 9:11am
Wait for Toast
9:11am 9:16am
9:11am 9:16am
Butter Toast
9:16am 9:20am
9:16am 9:20am
Leave for Lectures
9:30am 9:30am
9:30am 9:30am

Gantt Chart

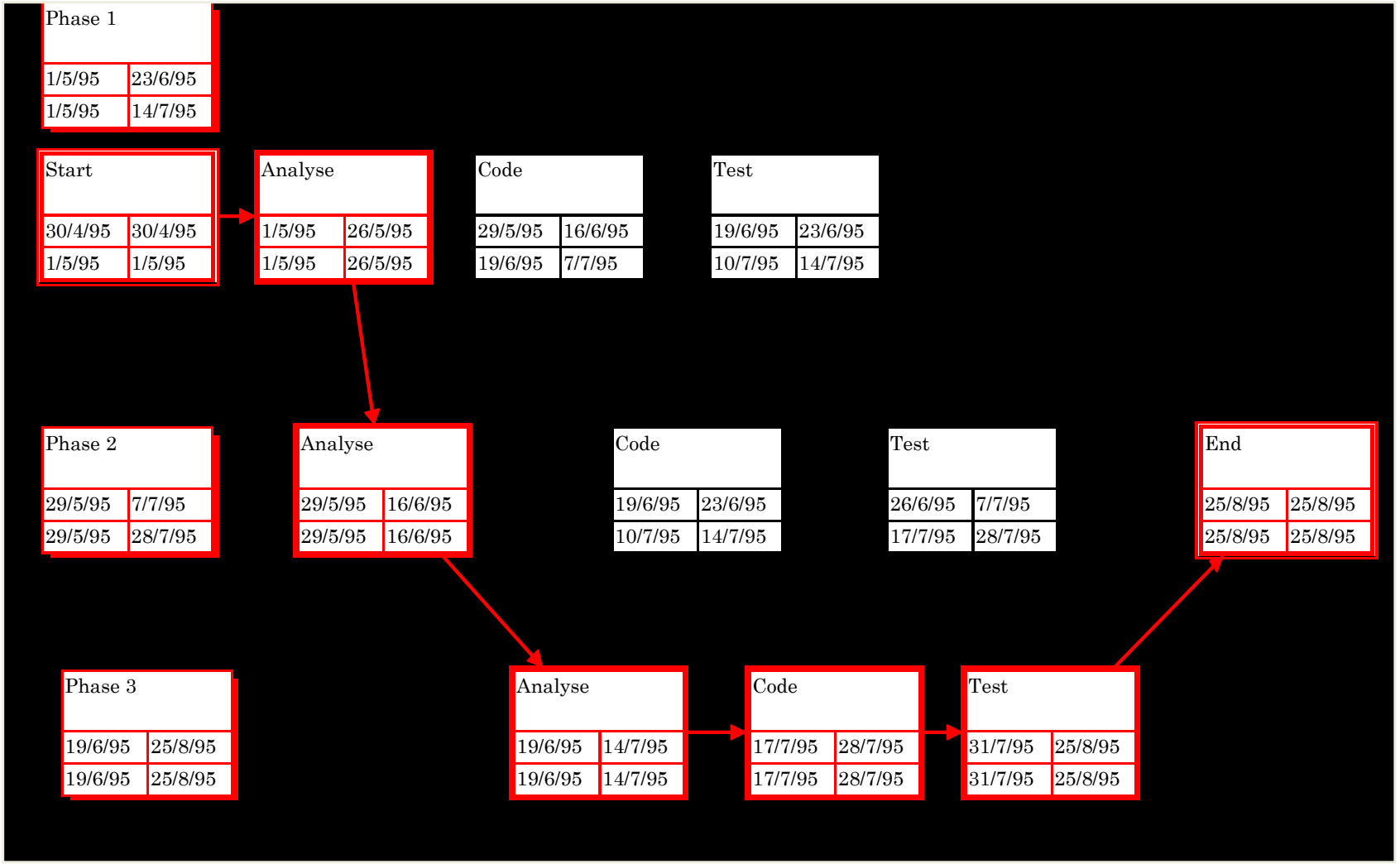


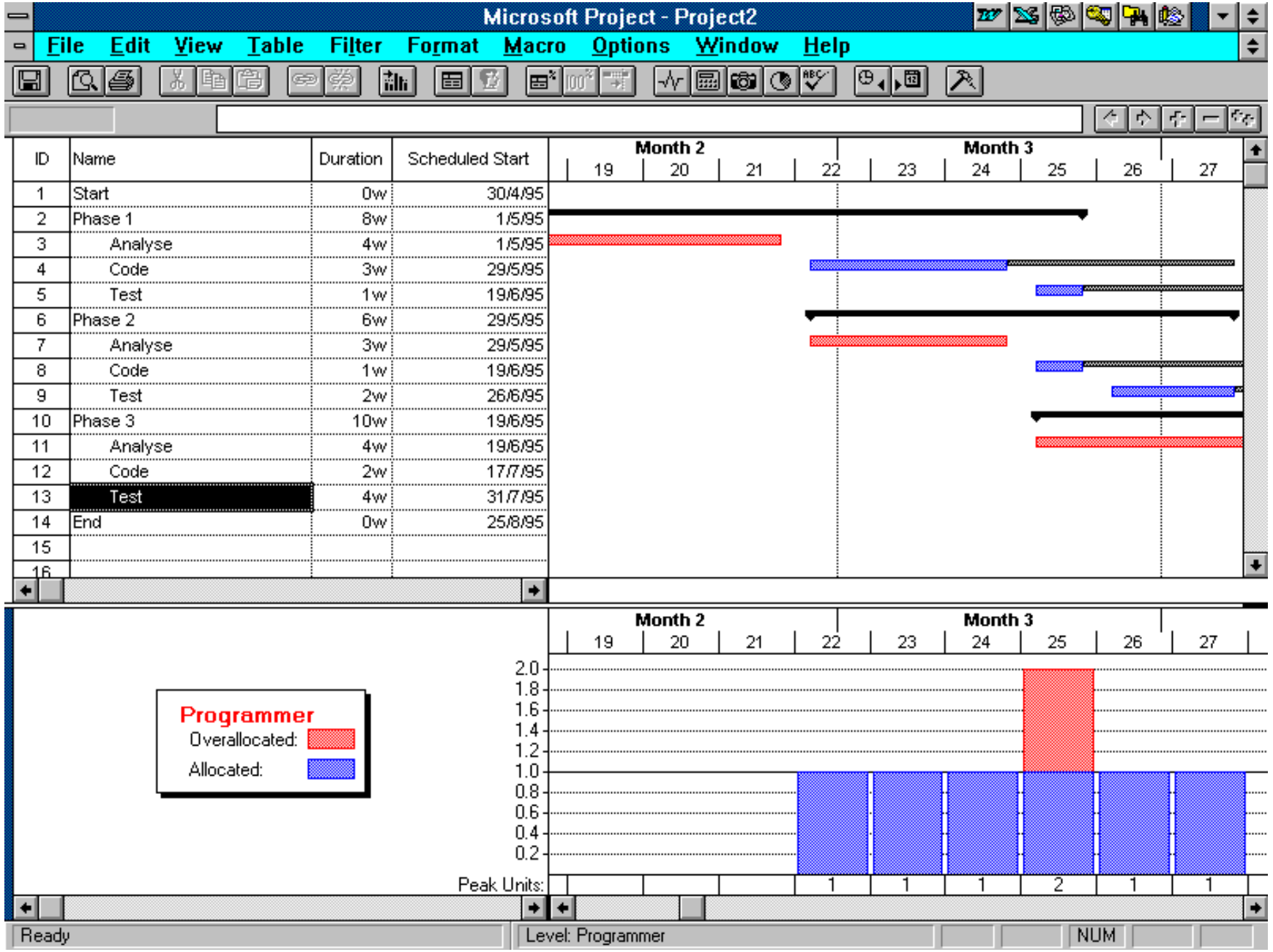
ID	Name	Duration
1	Get out of bed	0m
2	Wash	5m
3	Get dressed	5m
4	Put Kettle on	3m
5	Wait for kettle to boil	5m
6	Make Coffee	3m
7	Put toast on	5m
8	Wait for Toast	5m
9	Butter Toast	4m
10	Eat Breakfast	10m
11	Leave for Lectures	0m

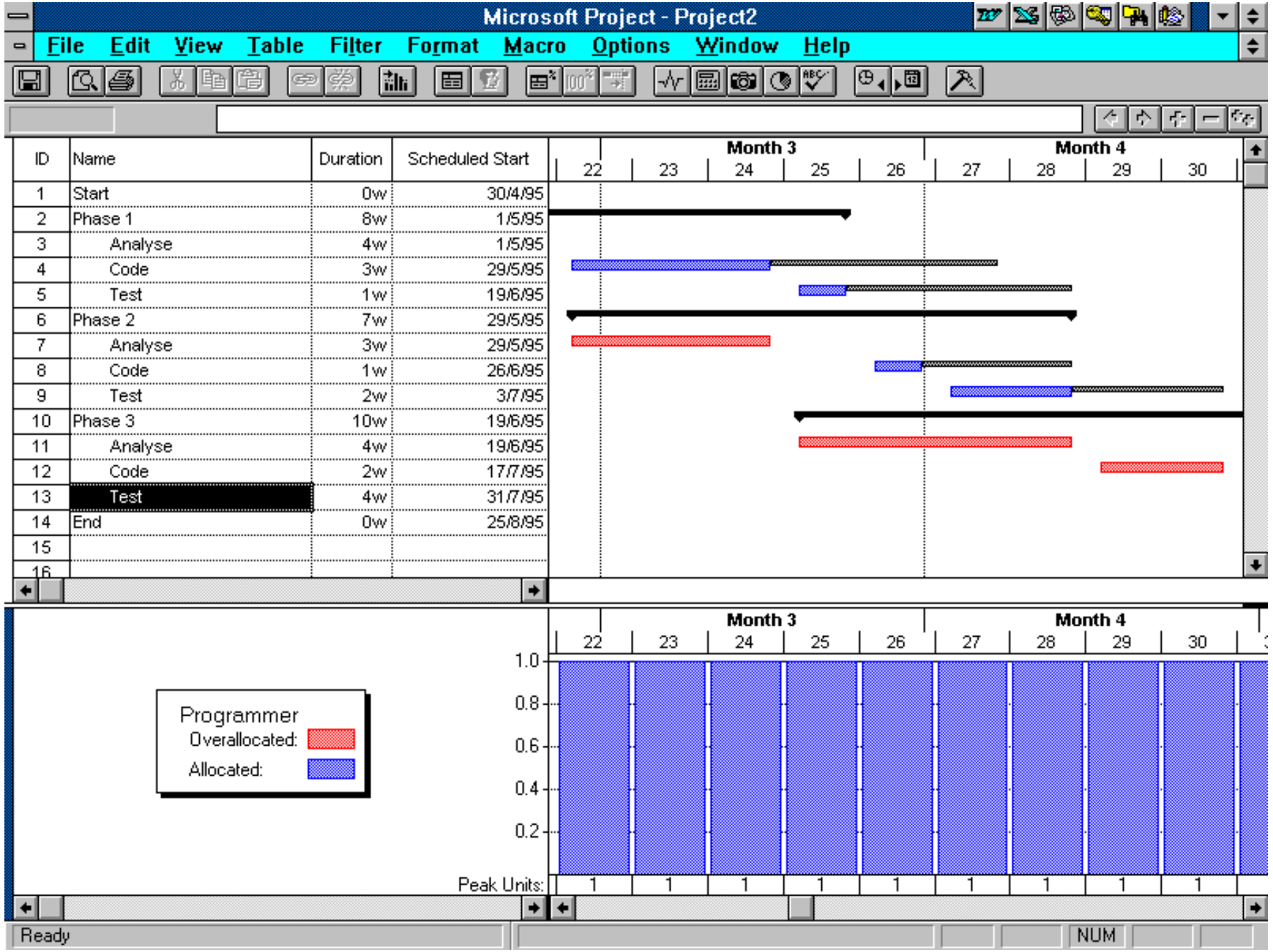
Example


Month 2
Month 3
Month 4
Month 5
ID
Name
Durat
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
1 Start 0v
2 Phase 1 8v
3 Analyse 4v
4 Code 3v
5 Test 1v
6 Phase 2 6v
7 Analyse 3v
8 Code 1v
9 Test 2v
10 Phase 3 10
11 Analyse 4v
12 Code 2v
13 Test 4v
14 End 0v

Example Pert



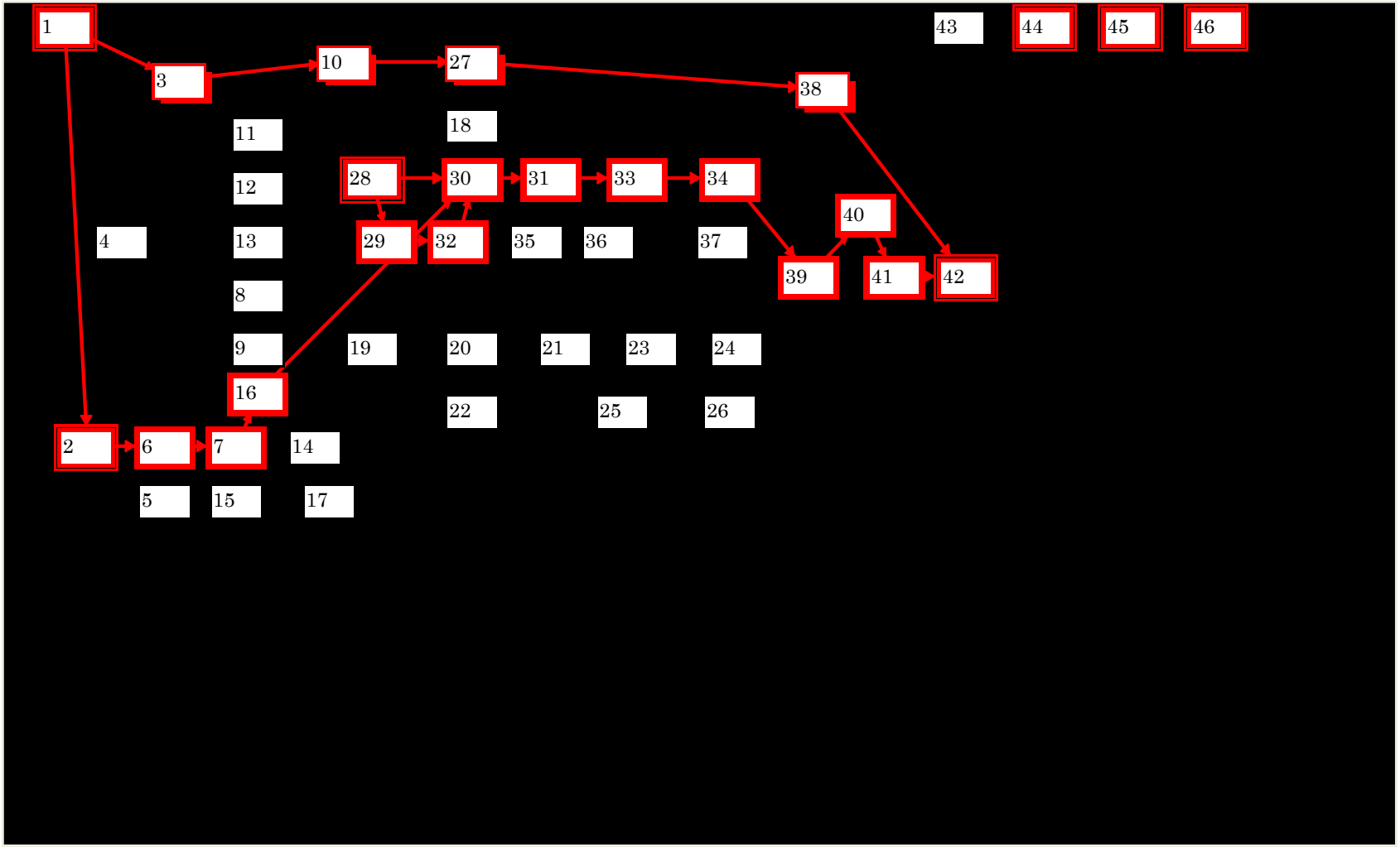

Microsoft Project - Project2
File Edit View Table Filter Format Macro Options Window Help
ID Name Duration Scheduled Start
1 Start 0w 30/4/95
2 Phase 1 8w 1/5/95
3 Analyse 4w 1/5/95
4 Code 3w 29/5/95
5 Test 1w 19/6/95
6 Phase 2 6w 29/5/95
7 Analyse 3w 29/5/95
8 Code 1w 19/6/95
9 Test 2w 26/6/95
10 Phase 3 10w 19/6/95
11 Analyse 4w 19/6/95
12 Code 2w 17/7/95
13 Test 4w 31/7/95
14 End 0w 25/8/95
15
16
Month 2
Month 3
19 20 21 22 23 24 25 26 27
Programmer
Overallocated:
Allocated:
2.0 1.8 1.6 1.4 1.2 1.0 0.8 0.6 0.4 0.2
Peak Units: 1 1 1 2 1 1
Ready
Level: Programmer
NUM


Microsoft Project - Project2
File Edit View Table Filter Format Macro Options Window Help
ID Name Duration Scheduled Start
1 Start 0w 30/4/95
2 Phase 1 8w 1/5/95
3 Analyse 4w 1/5/95
4 Code 3w 29/5/95
5 Test 1w 19/6/95
6 Phase 2 7w 29/5/95
7 Analyse 3w 29/5/95
8 Code 1w 26/6/95
9 Test 2w 3/7/95
10 Phase 3 10w 19/6/95
11 Analyse 4w 19/6/95
12 Code 2w 17/7/95
13 Test 4w 31/7/95
14 End 0w 25/8/95
15
16
Month 3
Month 4
22 23 24 25 26 27 28 29 30
Programmer
Overallocated:
Allocated:
1.0
0.8
0.6
0.4
0.2
Peak Units:
1 1 1 1 1 1 1 1 1
Ready
NUM

Levelling

- Adjust tasks to match resources available
- Automatic systems available, but do not always give an optimum result
- Tasks may be delayed within slack without affecting project dates
- Otherwise consider extending project, or using more resource
- Adding resource to late project may cause *RECURSIVE COLLAPSE*
  - consider carefully whether the benefits outweigh the

Larger example



Estimation Techniques

- Experience
- Comparison with similar tasks
  - 20 lines of code/day
  - can vary by 2 orders of magnitude
- Decomposition
- Plan to throw one away
- 20 working days per month BUT 200 per year

Rules of Thumb

- Software projects:
  - estimate 10 x cost and 3 x time
- 1:3:10 rule
  - 1: cost of prototype
  - 3: cost of turning prototype into a product
  - 10: cost of sales and marketing
  - >>Product costs are dominated by cost of sales
- Hartree's Law
  - The time to completion of any project, as estimated by the project leader, is a

Cynic’s Project Stages

- Enthusiasm
- Disillusionment
- Panic
- Persecution of the innocent
- Praise of the bystander