The Mythical Man-Month and Other Human Factors

Seminar „Software Cost Estimation“

Irene Bonomo, 5th November, 2002

Introduction

- There is a tendency in the IT industry to concentrate on technology and neglect people, but...
  - ...personnel factors have the greatest single impact on the software engineering productivity, and
  - ...the major causes why projects fail are not so much technological as sociological in nature.

Introduction

Comments on the Books

- Stevenson, 1995: covers a lot of aspects of software development, well structured, rather academic
- Brooks, 1975 (anniversary edition 1995): collection of essays, based on the author‘s experience in managing OS projects, written for project managers, very famous, partly out-of-date
- DeMarco/Lister, 1987: collection of essays about people in projects, examples of traps you can fall into as a project manager, very entertaining, very popular, still relevant

Contents

- Estimation
- Staffing
- Individual Skills
- Social Environment
- Office Environment
- Conclusion

Why are Projects Late?

- Estimating techniques are poorly developed
- Estimating techniques confuse effort with progress (hiding the assumption that man and months are interchangeable)
- Schedule process is poorly monitored
- Adding manpower to a late project is a natural and traditional reaction
- Unjustified natural optimism („all will go well")

Estimation

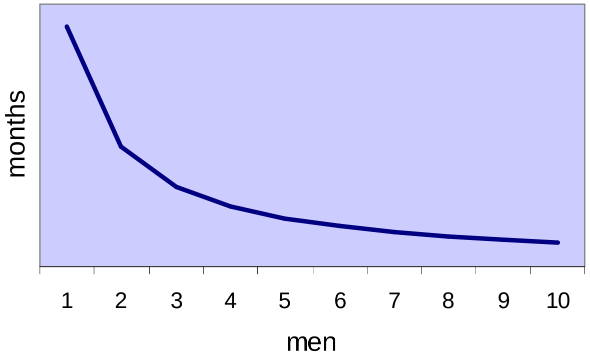
„Good cooking takes time. If you are made to wait, it is to serve you better, and to please you."

Antoine's chef

The Mythical Man-Month

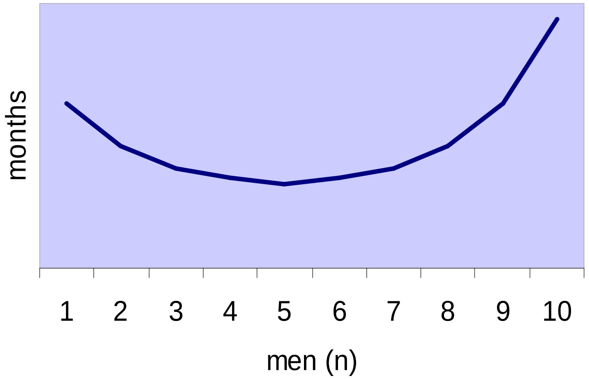
- Man-month is the unit of effort used in estimating and scheduling.
- Cost varies as the product of the number of men and the number of months. Progress does not.
- The man-month as a unit for measuring the size of a job is a dangerous and deceptive myth. It implies that men and months are interchangeable.
- Men and months are interchangeable only when a task can be partitioned among many workers with no communication among them.

Perfectly Partitionable Task



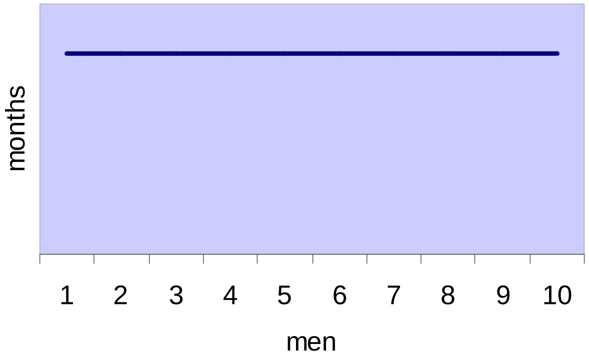
- e.g. reaping wheat, picking cotton

Task with Complex Interrelationships



- training effort increases linearly
- intercommunication effort increases as n (n-1) / 2

Unpartitionable Task



- e.g. bearing of a child

Conclusion

Adding manpower to a late project makes it later!

Parkinson's Law Revisited

- Project survey determining the productivity effect of various estimating methods:

Effort Estimate Prepared by	Average Productivity
programmer alone	8.0
supervisor alone	6.6
programmer and supervisor	7.8
systems analyst	9.5
no estimate	12.0

- Projects for which no estimates were prepared at all far outperformed all the others.

General Shortage of Skilled People

- There is an abundance of would-be programmers, but a general shortage of skilled people
- Causes:
  - growth of the industry, low productivity, lack of training, recruitment, economic recessions, poor deployment of staff, people leaving the industry, turnover
- Cures:
  - increase productivity, improve deployment, reduce turnover, provide training

Staffing

Turnover

- Magnitude:
  - 5 – 90% p.a. depending on different factors; decreased for the last decades
- Effects:
  - interchange of knowledge, loss of experience and efficiency, recruitment and induction costs (equivalent of one person-year of work), chain effect
- Causes:
  - shortage, lacking career-path, large organizations, pressure, desire for personal growth, personal conflicts, salary, location, lack of creativity and autonomy, mergers and takeovers
- Cures:
  - recruitment, training, career development, job satisfaction, personal growth

Recruitment

Circus Manager: How long have you been juggling?
Juggler: Oh, about six years.

Circus Manager: Can you handle three balls, four balls, and five balls?
Juggler: Yes, yes, and yes.

Circus Manager: Do you work with flaming objects?
Juggler: Sure.

Circus Manager: …knives, axes, open cigar boxes, floppy hats?
Juggler: I can juggle anything.

Circus Manager: Do you have a line of funny patter that goes with your juggling?
Juggler: It's hilarious.

Circus Manager: Well, that sounds fine. I guess you'r hired.
Juggler: Umm… Don't' you want to see me juggle?
Circus Manager: Gee, I never thought of that.

Training I

- Training is important because of the rapid evolution of the computer field
- Benefits:
  - fewer mistakes and omissions, more accurate estimations, higher productivity, better quality products, better service to customers
- Training should be considered as an investment producing a measurable return (like research and development)
- **Training is expensive, but no training is even more expensive**

Individual Skills

Training II

- Amount:
  - 10 – 15 days per year per person
- Cost:
  - about 2% of the IT budget containing salaries, travel and accommodation of the training staff, training facilities, opportunity cost, development of training materials
- Methods:
  - on-the-job, classroom, computer-based (CBT), hands-on

Experience and Ability I

- A person‘s performance depends on his innate ability (potential), and his knowledge (training, experience)
- Productivity weighting factors (Aron, 1974):

senior programmer:	0.5
programmer:	1.0
apprentice programmer:	1.5
trainee programmer:	3.0

- Experienced programmers make less errors, write more compact code, are better in removing bugs

Social Environment

Experience and Ability II

- Principle of top talent: Use better and fewer people
  - the top 20% of programmers produce 50% of the output, while the bottom 50% of them only produce 20%.
  - senior programmers are only paid twice as much as juniors, and are therefore not paid what they are worth.
- Types of ability: analytical mind to find errors, synthetical mind to fix them, considerable persistence to find errors, ability to write for documentation purposes, communication ability for working in teams
- Are good programmers made or born ?

Jelled Teams

- Group of people so strongly knit that the whole is greater than the sum of the parts
- **The purpose of a team is not goal attainment but goal alignment**
- Characteristic signs of jelled teams:
  - low turnover
  - strong sense of identity
  - sense of eliteness
  - joint ownership of the product
  - obvious enjoyment

Teamicide

- You can‘t easily "make" teams jell, you just can hope they will jell, ...
- …but you can do a lot to prevent them from jelling by committing „teamicide“:
  - □ defensive management
  - □ bureaucracy
  - □ physical separation
  - □ fragmentation of time
  - □ quality reduction
  - □ unreachable deadlines
  - □ clique control

Management I

In my early years as a developer, I was privileged to work on a project managed by Sharon Weinberg (…) She was a walking example of much of what I now think of as enlightened management. One snowy day, I dragged myself out of a sickbed to pull together our shaky system for a user demo. Sharon came in and found me propped up at the console. She disappeared and came back a few minutes later with a container of soup. After she'd poured it into me and buoyed my spirits, I asked her how she found time for such things with all the management work she had to to. She gave me her patented grin and said, "Tom, this *is* management."

Tom De Marco

Management II

One of my first bosses was Jerry Wiener (…) At the time I came along, the company was about to enter into a contract that was larger than anything it had ever done before. The entire staff was assembled as our corporate lawyer handed Jerry the contract and told him to read it and sign on the last page. "I don't read contracts," Jerry said, and started to sign. "Oh, wait a minute," said the lawyer, "let me go over it one more time."

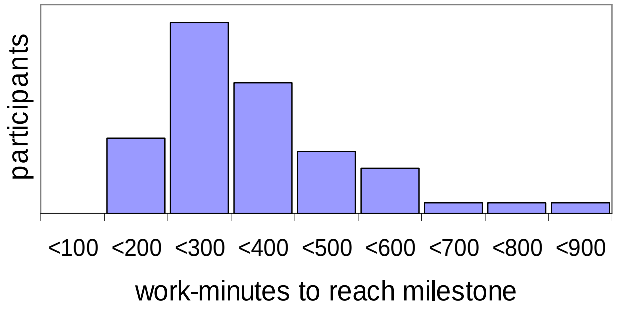
Tom De Marco

Great Managers...

- ...make a cult of quality.
- ...provide lots of feelings of success.
- ...build a sense of eliteness and uniqueness.
- ...allow and encourage heterogeneity.
- ...preserve and protect successful teams. They never change a winning team!
- ...consider team structures as networks and not as hierarchies.
- ...provide strategical but not tactical direction.

Office Environment

Coding War Games I



Coding War Games II

Environmental Factor	Those Who Performed in 1st Quartile	Those Who Performed in 4th Quartile
How much dedicated workspace do you have?	7 m^2	4.1 m^2
Is it acceptably quiet?	57% yes	29% yes
Is it acceptably private?	62% yes	19% yes
Can you silence your phone?	52% yes	10% yes
Can you divert your calls?	76% yes	19% yes
Do people often interrupt you needlessly?	38% yes	76% yes

Brain Time vs. Body Time

- Workers divide their time as follows:

Work Mode	Percent of Time
working alone	30%
working with other people	50%
working with two or more people	20%

- The *Environmental Factor* or is a meaningful metric of how good or bad your environment is:

E-Factor = Uninterrupted Hours / Body-Present Hours

The Ideal Office Environment...

- ...achieves the most efficient flow of work and overcomes bottlenecks (e.g. printers),
- ...increases productivity,
- ...enhances employee satisfaction and well-being,
- ...minimizes health and safety risks (e.g. radiation), and
- ...is free from distractions and interruptions (specially for software engineers).

Conclusion

Influence on Productivity

- Physical environment:
  - □ lighting, air temperature, noise, humidity, ventilation
- Office layout:
  - □ open-plan vs. enclosed office
- Furniture:
  - □ chair, desk
- Workstation
  - □ display, keyboard, mouse
- Office Automation

Conclusion

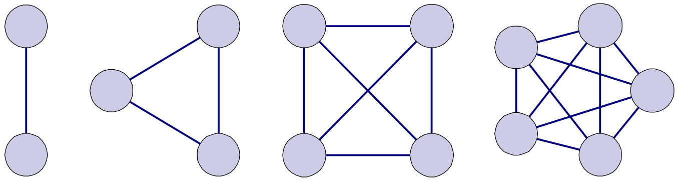
- Focussing on people and their needs can have a considerable positive effect on productivity and project success
- Why are human factors still neglected in the IT industry?
  - □ it's easier to solve technical problems
  - □ weak managers feel threatened by strong workers
  - □ short-term costs vs. long-term benefits

Reserve

Scheduling a Software Task

- Testing is the most mis-scheduled part of programming.
- Delay at this point has unusually severe financial and psychological effects.
- Rule of thumb:
  - □ 1/3 planning
  - □ 1/6 coding
  - □ 1/4 component test
  - □ 1/4 system test

Intercommunication



men:	2	3	4	5
connections:	1	3	6	10

Some Productivity Gain Potential in Detail

- QWERTY keyboard
  - □ originally designed to *prevent* fast typing, so that the mechanism would not jam!
  - □ only 32% of keystrokes on the „home row“
  - □ alternatives (e.g. Dvorak keyboard) not generally accepted yet
- one-finger vs. touch-typing technique
  - □ 25 – 100% improvement in typing speed ...
  - □ ... and 75% increase in accuracy by typing speed training
- telephone calls
  - □ 11 – 17% of the working time spent on the telephone
  - □ less than 50% of the telephone calls are successful
  - □ To be replaced by voice mail or electronic mail