

an introduction to:

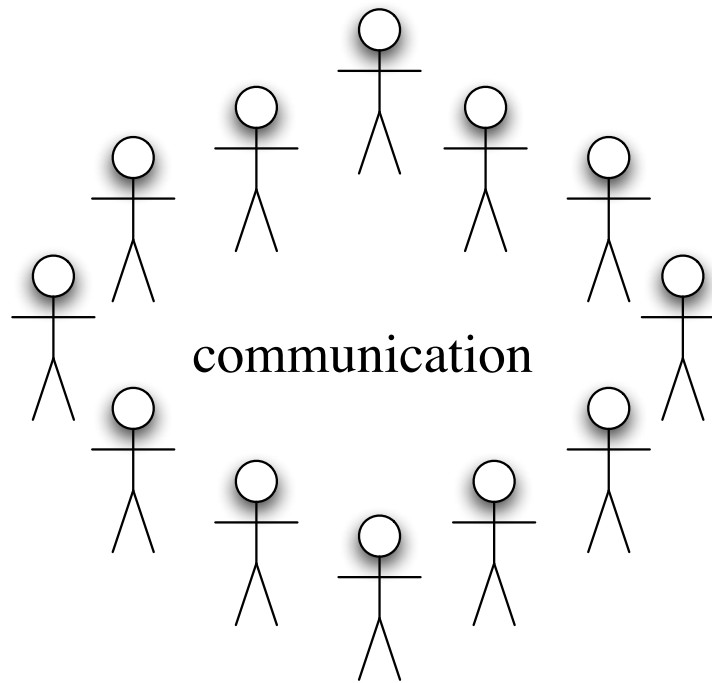
peopleware

tom demarco & timothy lister

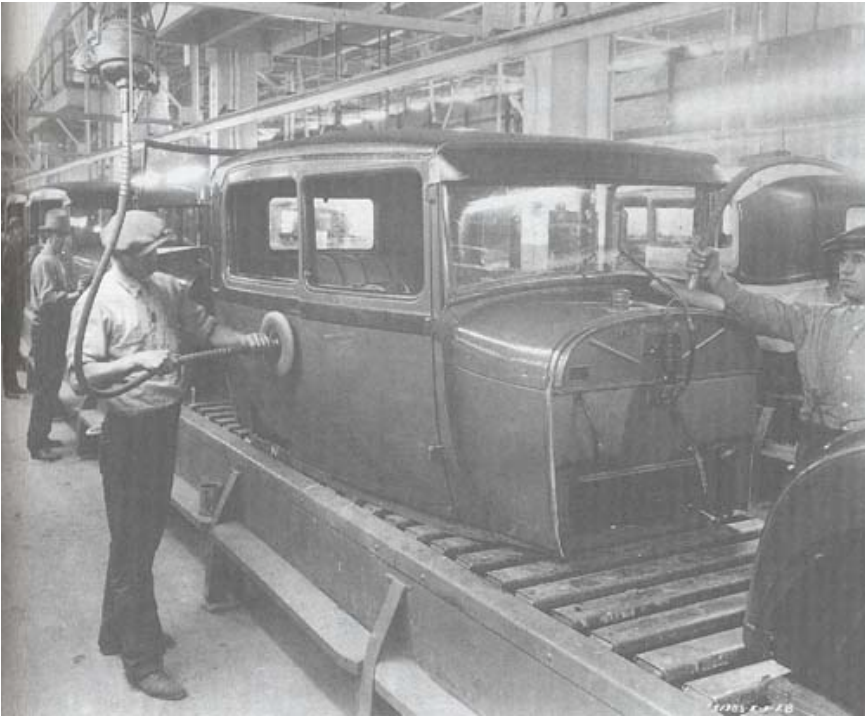
“the major problems of our work are not so much **technological** as **sociological** in nature.” [p. 4]

managing people

**thinking about managing software
work**

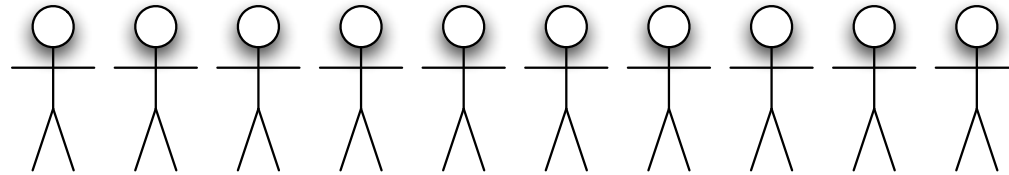


“because we go about this work in teams and projects and other tightly knit working groups, we are mostly in the human communication business. our successes stem from good human interactions... and our failures stem from poor human interactions.” [p. 5]



management: the bozo definition
“management is kicking ass” [p. 9]

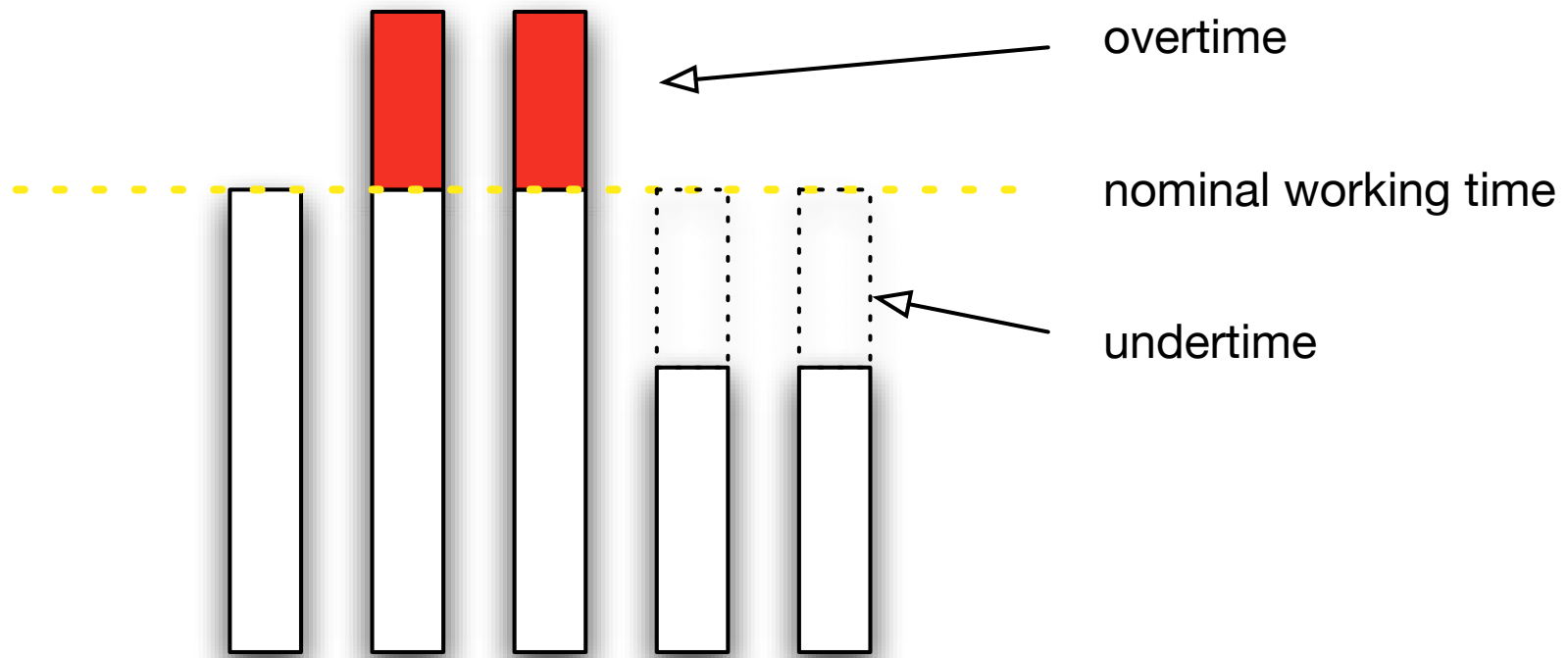
the people store

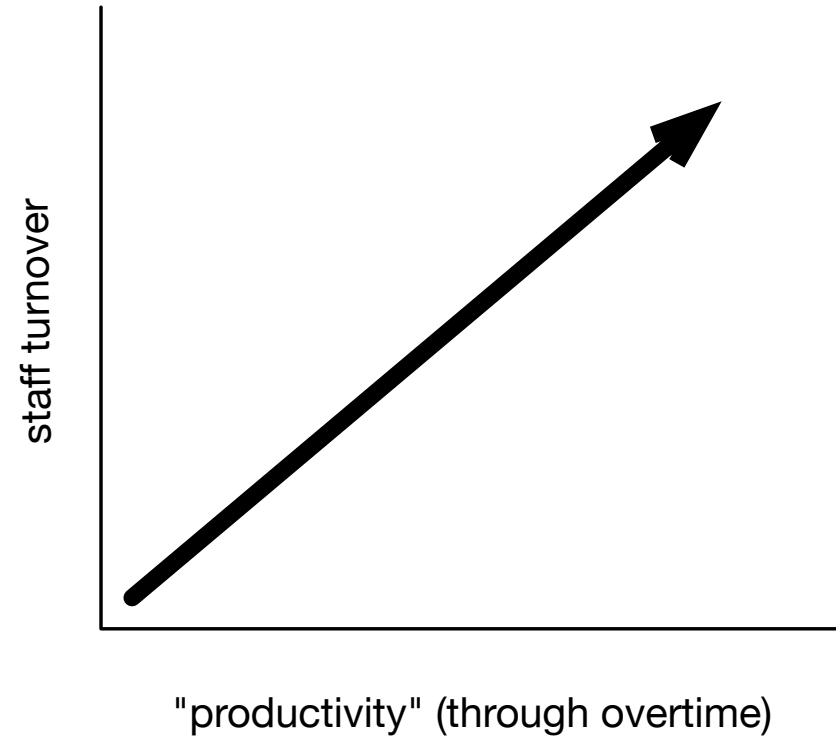


people are not the same! therefore they are not
interchangeable

“the manager’s function is not to make people work but to make it possible for people to work” [p. 34]

overtime



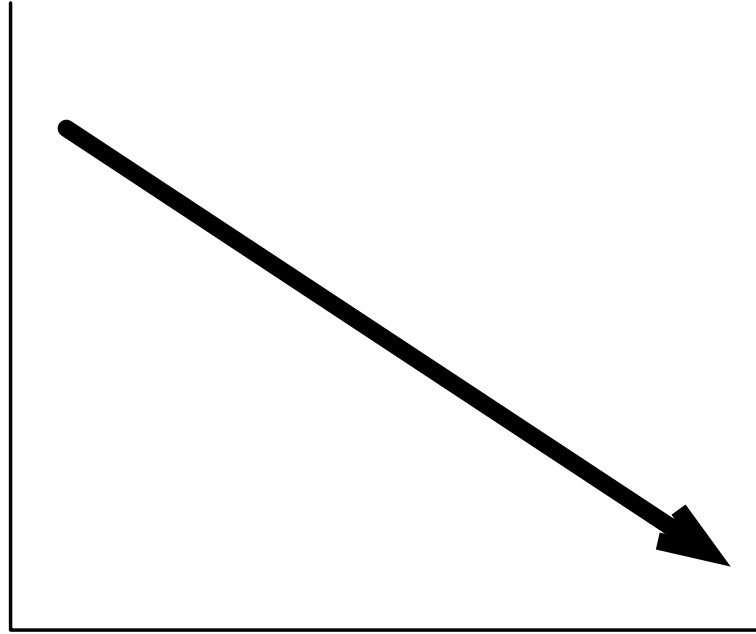


“people under pressure don’t work better; they just work faster” [p. 18]

quality - if time permits

why does quality matter to workers?

quality



deadline pressure

who's to blame?

it's all the customer's fault - they
accept poor-quality products [p. 21]

“quality is free, but only to those who are willing to pay heavily for it.” [p. 23]

productivity

effort estimated by	average productivity	number of projects
programmer alone	8.0	19
supervisor alone	6.6	23
programmer & supervisor	7.8	16
systems analyst	9.5	21
no estimate	12.0	24

“organizational busy work tends to expand to fill the working day.” [p. 29]

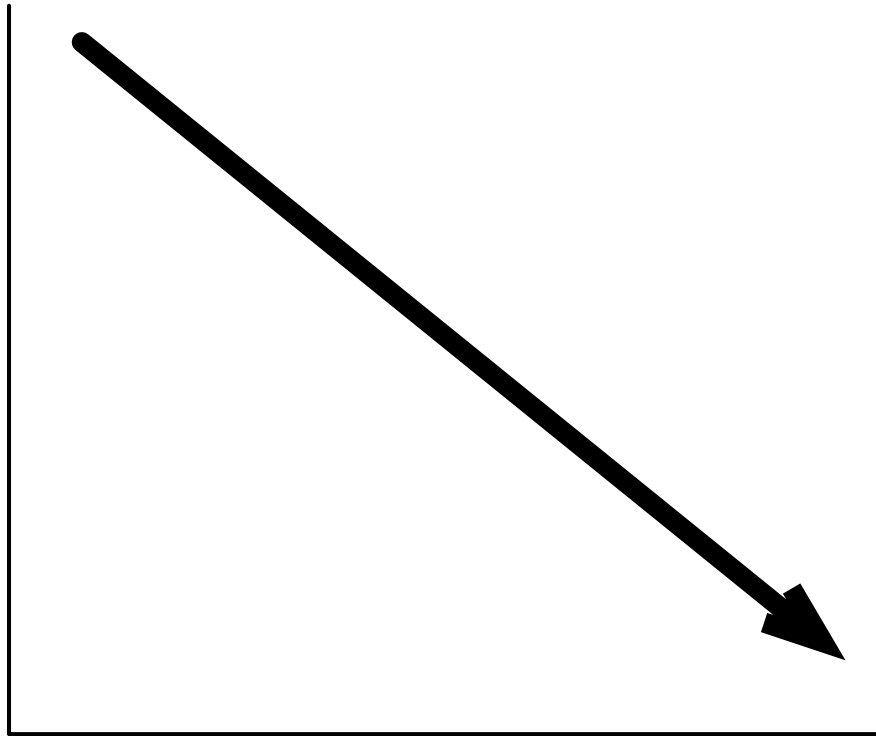
managing people:

- thinking about managing people and software work
- overtime
- quality - if time permits
- productivity

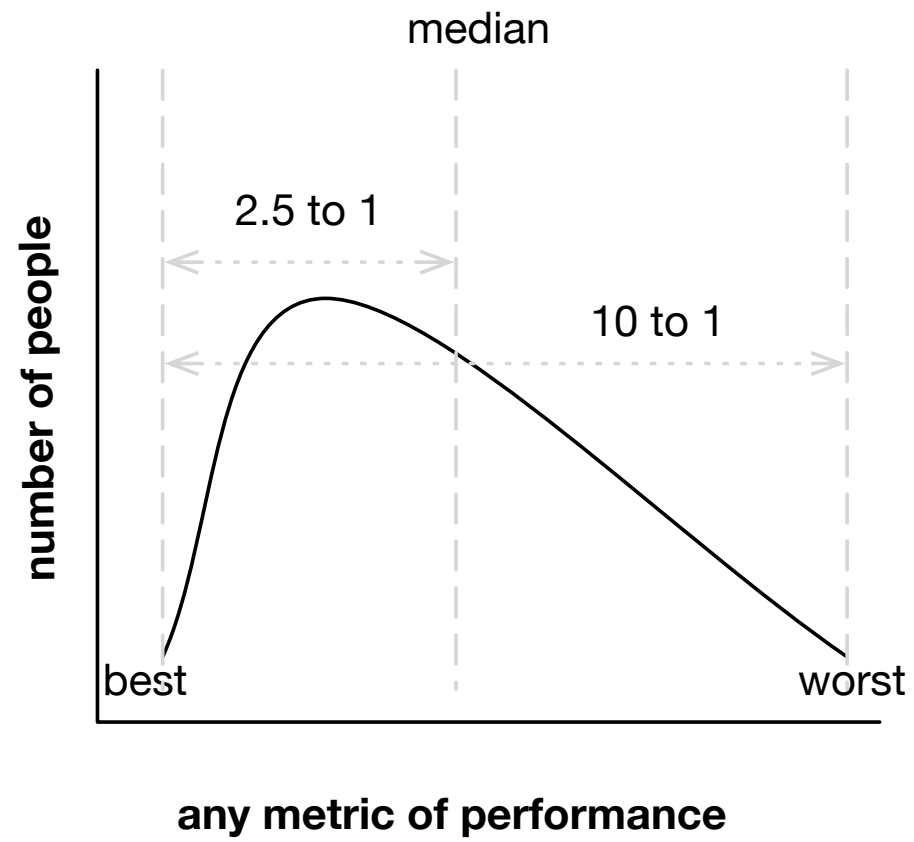
the office environment

environment and productivity

concentration



noise & interruptions



little or no correlation with performance:

- coding language (except for assembly!)
- years of experience (except for coders with less than 6 months experience with language)
- number of defects (on average, coders with zero defects were **faster**)
- salary

but:

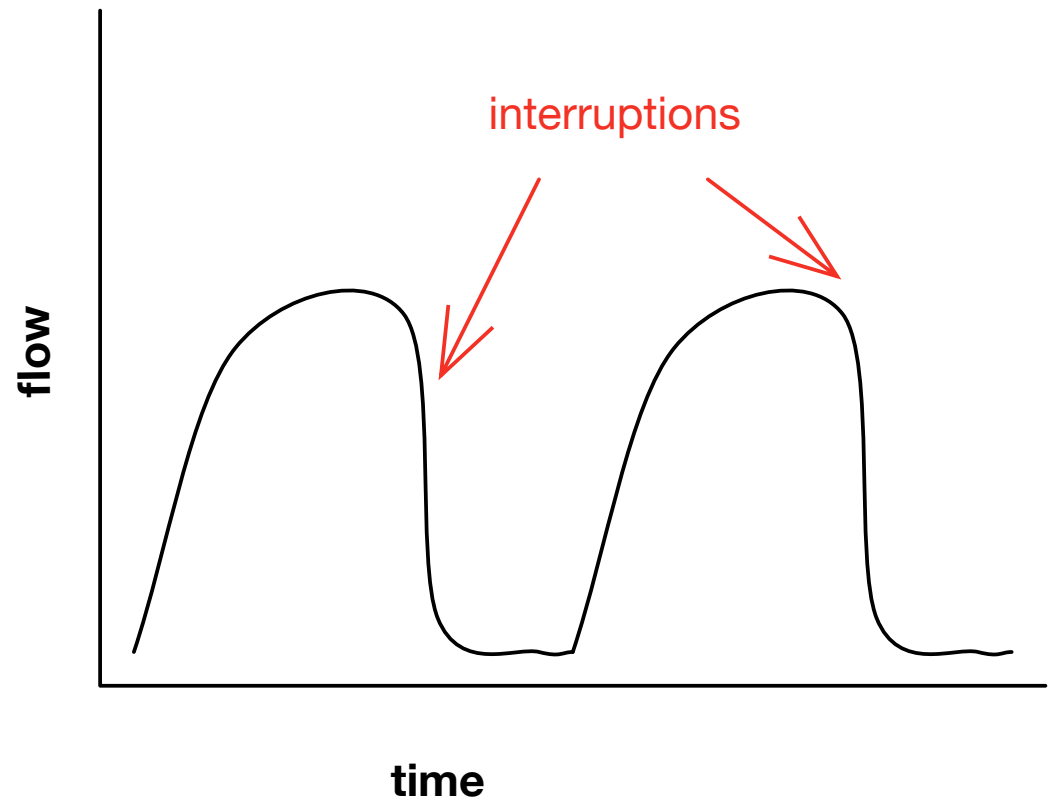
coders from same organisation had
very similar performance

high correlation with performance:

- office space per developer
- acceptable levels of quietness
- privacy
- fewer interruptions

**how software developers spend their
time**

mode	percentage of time
working alone	30%
working with one other person	50%
working with two or more other people	20%



patterns (the original)

office environment:

- environment and productivity
- how developers spend their time
- flow

the right people

the right people are fundamental

hiring the right people

turnover

typical turnover figures:
30% to 80%

organisations with low turnover “share
a preoccupation with being the
best” [p. 111]

Methodology with a capital M

effects of Methodology:

- a morass of paperwork
- a paucity of methods
- an absence of responsibility
- a general loss of motivation

the right people:

- the right people are fundamental
- hiring the right people
- staff turnover
- Methodology with a capital M

**growing productive
teams**

gel



management by hysterical optimism

signs of a gelled team:

- low turnover during the project
- a strong sense of team identity
- a sense of eliteness
- joint ownership of the product
- obvious enjoyment of the work

how to stop a team from gelling:

- defensive management (mistrust)
- bureaucracy
- physical separation
- fragmentation of people's time
- reduce product quality
- phony deadlines
- clique control

“most organisations don’t set out
consciously to kill teams. they just act
that way.” [p. 4]

chemistry for team formation:

- make a cult of quality
- provide lots of satisfying closure
- build a sense of eliteness
- allow and encourage heterogeneity
- preserve and protect successful teams
- provide strategic but not tactical direction

growing productive teams:

- gel
- management by hysterical optimism
- signs of a gelled team
- how to stop teams from gelling
- chemistry for team formation