



CS 428

Why People and Teams Matter

WINTER 2020, WEEK #2

BRUCE F. WEBSTER

- ▶ Study and experienced observers agree: people matter most in affecting regarding IT project success or failure
- ▶ Core issue: a lot of that is natural talent (cf. music, math, art, athletics), and not that many people are inherently good at IT
- ▶ Over the past 70 years, demand for IT personnel had massively outstripped supply
- ▶ That trend will continue
- ▶ Observations and experiences?

WEB #1: "The Real Software Crisis"
(BYTE, 1996)

- ▶ Follow-up to “Real Software Crisis” – identifies five essential qualities to look for in hiring IT personnel
 - ▶ **Talent:** inherent IT-related talents (not a single talent, not a single yes/no)
 - ▶ **Experience:** our most painful experiences are usually our most informative
 - ▶ **Professionalism:** be reliable, be reasonable, get your work done
 - ▶ **Education:** those with education in CS/IT have an advantage over those without
 - ▶ **Skills:** honed skills in specific languages, methodologies, technologies
- ▶ Observations and experience?

WEB #1: “TEPES” (2008)

- ▶ In dysfunctional IT organizations, your best people tend to depart quickly, leaving behind the less talented, less skilled, less competent
- ▶ Overall quality of IT organization declines over time and becomes hard to improve
- ▶ It's not unique to IT (though IT is very sensitive to it)
- ▶ It's not true of all IT organizations; just dysfunctional ones or those heading towards dysfunction
- ▶ Not everyone left behind is necessarily incompetent
- ▶ Observations and experience?

WEB #1: "The Dead Sea Effect" (2008)

- ▶ Wide variation in programming productivity (up to 10:1)
 - ▶ Keep the best programmers and fire everyone else
- ▶ Problem: the small, sharp team is too small for really large projects
- ▶ Surgical team approach: provide support to the most productive
- ▶ Focuses on there being a 'chief surgeon' to make final decisions
 - ▶ Could be a chief architect, or could be a chief programmer
- ▶ Comments and observations?

MMM: Ch 3: The Surgical Team

- ▶ Management tends to see development like making fast food
- ▶ Tendency to punish/bury errors and dead ends (cf. Armour)
- ▶ Common attitude that “management provide[s] all the thinking and the people underneath just carry out their bidding.”
- ▶ Common attitude that people are interchangeable parts and that interrelations – good or bad – aren’t important
- ▶ Steven Covey: we are often too busy sawing to sharpen the saw
- ▶ PW: “The average software developer...doesn’t own a single book on the subject of his or her work and hasn’t ever read one.” [Hence this class]
- ▶ Observations and feedback?

PW Ch 2: Make a Cheeseburger, Sell a Cheeseburger

- ▶ “Real-world” management too often “is all about getting people to work harder and longer, largely at the expense of their personal lives.”
 - ▶ There ain’t no such thing as [sustained] overtime
 - ▶ Remember, too: no other success can compensate for failure in the home
 - ▶ Billboard on I-15: “You can code and still go home to your family at night.”
- ▶ Such management is often blind to costs of turnover and burnout
- ▶ People under time pressure don’t work better – just “faster”
 - ▶ Result is poorer quality and more job dissatisfaction
- ▶ Observations and feedback?

PW Ch 3: Vienna Waits for You

Pathological (power-oriented)	Bureaucratic (rule-oriented)	Generative (performance)
Low cooperation	Modest cooperation	High cooperation
Messengers "shot"	Messengers neglected	Messengers trained
Responsibilities shirked	Narrow responsibilities	Risks are shared
Bridging discouraged	Bridging tolerated	Bridging encouraged
Failure leads to scapegoating	Failure leads to justice	Failure leads to inquiry
Novelty crushed	Novelty leads to problems	Novelty implemented

ACC #3: Westrum Organizational Culture Models (Background)

- ▶ Focus on Lean Management practices
 - ▶ Lean Management => Westrum Org / Delivery Performance / Less Burnout
- ▶ Combination of limiting Work in Progress (WIP) *and* Visual Management is critical
 - ▶ Limit WIP: take small bites, not big ones
 - ▶ Visual Management: physical and/or digital dashboards to show key metrics and work status
- ▶ Implement lightweight change management process
 - ▶ Compared four approaches (all external, high-risk external, peer review, none)
 - ▶ Highest delivery performance: peer review or none
 - ▶ “Use a lightweight change approval process based on peer review, such as pair programming or intrateam code review, combined with a deployment pipeline to detect and reject bad changes”
 - ▶ Me: that last item sounds like a bit of hand-waving – unclear how it works in two-party setting

ACC Chapter 7: Management Practices for Software