Fall 2018

Bruce F. Webster

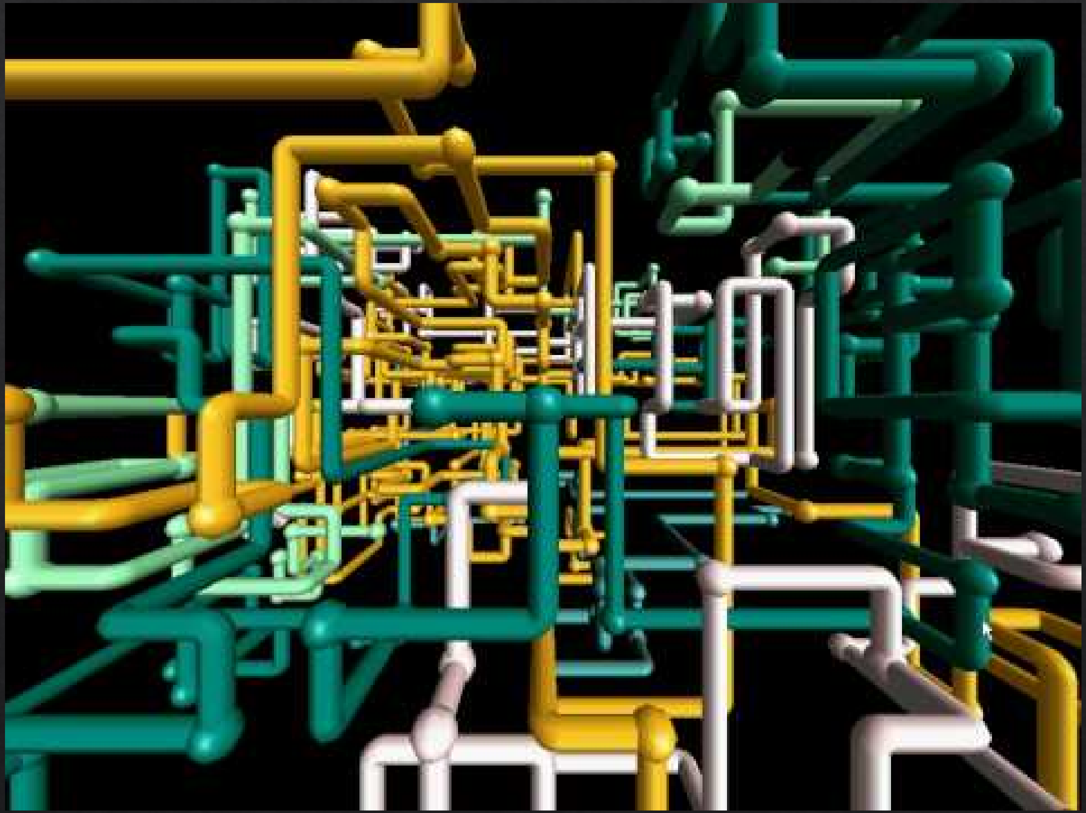
CS 428 – Webster readings #2

“How to retain IT talent with Goal Alignment” (Baseline, 2008) [Link]

- Trying to solve team issues at Pages Software – did a team offsite
  - Each engineer explained what her/his goals were for being at Pages
  - As a team, we crafted team goals that supported individual goals
  - We then determined how those team goals would support company goals
- Result: zero (0) voluntary turnover in engineering staff over 4.5 year period
- Problem: upper management tends to see engineers are interchangeable and subject to simple motivations
  - “Don’t they realize they’re dealing with grown-ups?”
- Observations and experiences?

“Remember Conways Law” (2013) [Link]

- Experience reviewing massive ($500M) failing IT project at Fortune 50 corp



“Remember Conway’s Law” (cont.)

- Coined by Fred Brooks in *The Mythical Man-Month*:
  - Any organization that designs a system (defined broadly) will produce a design whose structure is a copy of the organization’s communication structure.
- Put simply, architecture tends to follow organization, not the other way around
- Thus, you need to make sure your organization reflects your anticipated architecture (hint: you may end up revising your org charts)
- Observations and experience?

"Controlling IT costs: Using a Maintenance Architect" (Baseline, 2008) [Link]

- Problem: costs and difficulties of maintaining existing systems (50% to 80%)
  - Maintenance often used for entry-level personnel and old-timers
  - Software entropy sets in
- Possible solution: appoint a maintenance architect
  - Learn (and document) essential architecture of all production systems
  - Review all proposed changes to any given system (bug fix, enhancement, replacement)
  - Issue 'environmental impact statement' on consequences of such proposals
  - Oversee actual work on existing systems
  - Great training to become a chief software architect and/or CTO

“Negotiations and Lovesongs” (2008) [Link]

- Concept: applying n-player game theory to organizational software engineering
  - IT engineers (Geeks)
  - Management (Suits)
  - End-users (Users)
- The three groups have a hard time agreeing on what ‘game’ they’re playing, much less what ‘victory’ looks like
- Each group tends to have negative stereotypical views of the other two
- Result is a lot of mistrust and miscommunication