Evidence-based Scheduling

Michael Tsai

2017/5/23

Reference

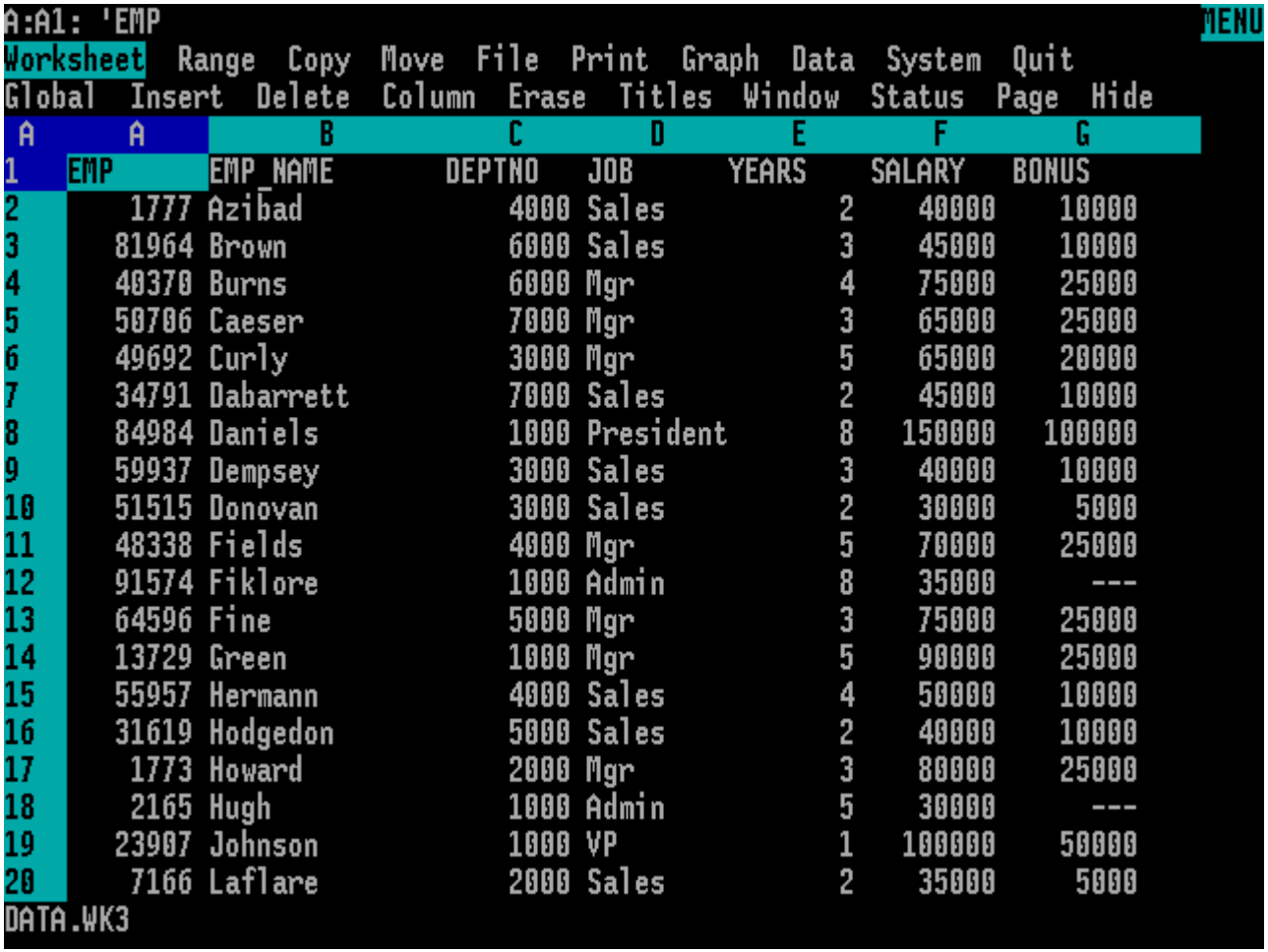
- http://www.joelonsoftware.com/items/2007/10/26.html

Why do you need a schedule?

- Bad example (no schedule):
  - Amtrak's Acela: express train from Boston to Washington, DC
    PR campaign starts before it finishes: 看得到吃不到
  - Lotus 1-2-3 v.s. Microsoft Excel
  - Netscape 5.0: 2 years late, 80%→ 20%
- Why? 'Cause the programmers do not want to make a schedule.
  - No use. Not realistic
  - Real pain in the ***

Amtrak's Acela

Lotus 1-2-3



A:A1: 'EMP MENU

Worksheet Range Copy Move File Print Graph Data System Quit

Global Insert Delete Column Erase Titles Window Status Page Hide

	A	B	C	D	E	F	G
1	EMP	EMP_NAME	DEPTNO	JOB	YEARS	SALARY	BONUS
2	1777	Azibad	4000	Sales	2	40000	10000
3	81964	Brown	6000	Sales	3	45000	10000
4	40370	Burns	6000	Mgr	4	75000	25000
5	50706	Caeser	7000	Mgr	3	65000	25000
6	49692	Curly	3000	Mgr	5	65000	20000
7	34791	Dabarrett	7000	Sales	2	45000	10000
8	84984	Daniels	1000	President	8	150000	100000
9	59937	Dempsey	3000	Sales	3	40000	10000
10	51515	Donovan	3000	Sales	2	30000	5000
11	48338	Fields	4000	Mgr	5	70000	25000
12	91574	Fiklore	1000	Admin	8	35000	---
13	64596	Fine	5000	Mgr	3	75000	25000
14	13729	Green	1000	Mgr	5	90000	25000
15	55957	Hermann	4000	Sales	4	50000	10000
16	31619	Hodgedon	5000	Sales	2	40000	10000
17	1773	Howard	2000	Mgr	3	80000	25000
18	2165	Hugh	1000	Admin	5	30000	---
19	23907	Johnson	1000	VP	1	100000	50000
20	7166	Laflare	2000	Sales	2	35000	5000

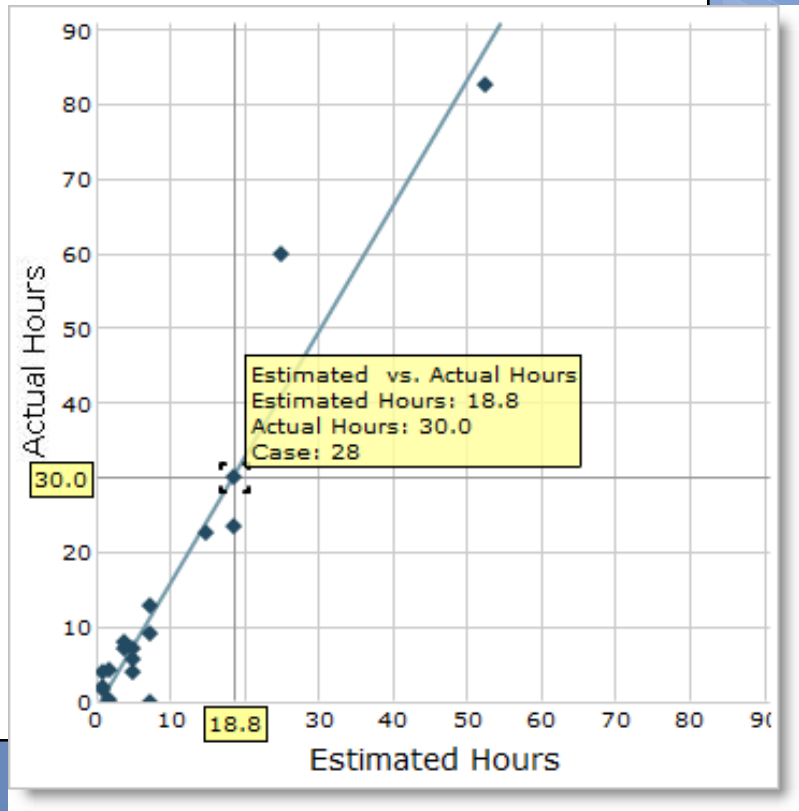
DATA.WK3

Break'er down

- Time unit: hours, not days (nothing more than 16 hrs)
- It forces you to figure out the details (start to design)
- (no details, then no way to figure out the time)

Track elapsed time

- Estimate is always wrong (almost):
  - Unpredictable bugs
  - Interruptions
  - (Hard drive/machine failure)
  - (火災警報)
  - (老師突然請導生宴)
- Know how much time you ACTUALLY spent on each task
- Get the *velocity (斜率)*



Track elapsed time

- The **velocity** can be:
  - 1.0 1.0 1.0 1.0 ….
  - 0.1, 0.5, 1.7, 0.2, 1.2, 0.9, 13.0, ….
  - 0.6, 0.5, 0.6, 0.6, 0.5, 0.6, 0.7, 0.6, ….

Simulate the Future

- Monte Carlo method: create 100 possible scenarios for the future.
  Each of these possible futures has 1% probability, so you can make a chart of the probability that you will ship by any given date.

Estimate:	4	8	2	8	16	
Random Velocity: (from history)	0.6	0.5	0.6	0.6	0.5	Total:
E/V:	6.7	16	3.3	13.3	32	71.3

  - 1.0 1.0 1.0 1.0 ….
  - 0.1, 0.5, 1.7, 0.2, 1.2, 0.9, 13.0, ….
  - 0.6, 0.5, 0.6, 0.6, 0.5, 0.6, 0.7, 0.6, ….
- Holidays, vacations, etc.

Obsessive-compulsive disorder not required

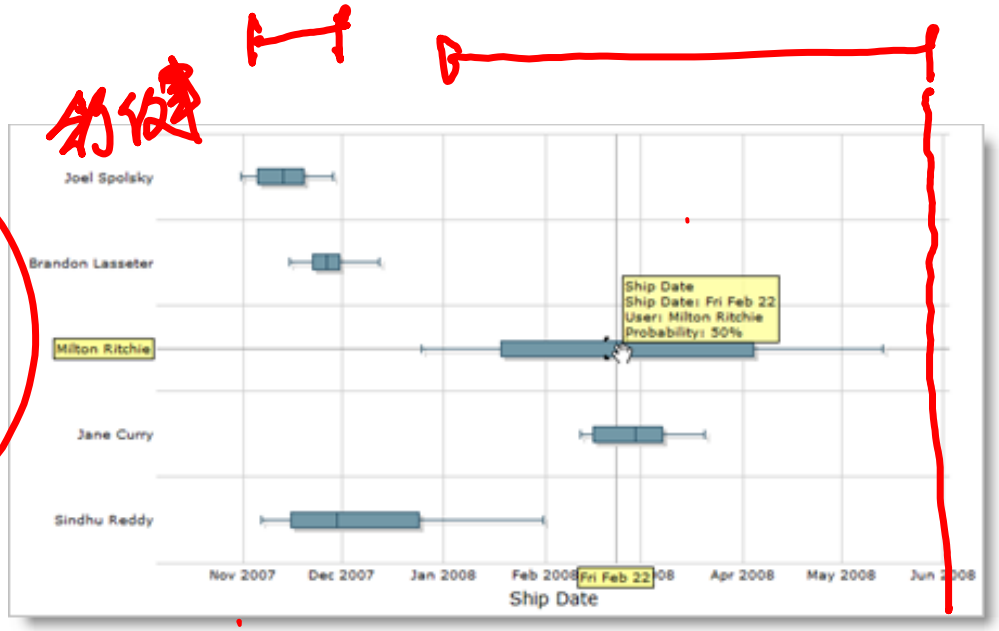
- How do we accounts for all sorts of interruptions?
- As long as we keep the clock running when we are interrupted, we are fine.
- Original estimates: {2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, … }
- Actual time with interruptions:{2, 2, 2, 2, 4, 2, 2, 2, 2, 4, 2, … }
- Velocities: {1, 1, 1, 1, 0.5, 1, 1, 1, 1, 0.5, 1, … }
- Monte Carlo simulations take the probability of interruptions into account!

2 types of programmers

- When developers get interrupted, they can either
  - make a big stink about putting the interruption on their timesheet and in their estimates, so management can see just how much time is being wasted on fishing conversation, or
  - make a big stink about refusing to put it on their timesheet, just letting the feature they were working on slip, because they refuse to pad *their* estimates which were *perfectly correct* with stupid conversation about fishing expeditions to which they *weren't even invited*.
- EBS gives the same results.

Manage your projects actively

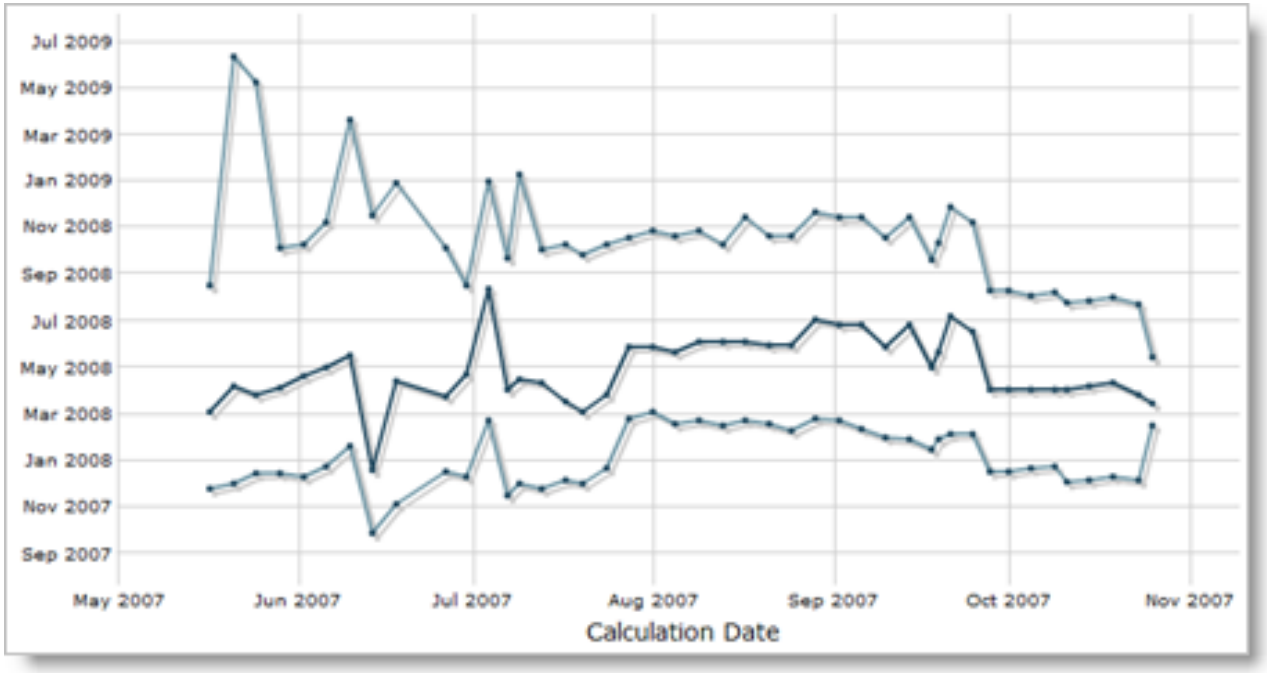
Priority	50% Date
1 – Urgent	11/30/2007
2 – High	12/15/2007
3 – Important	3/6/2008
4 – Medium	3/14/2008
5 – Moderate	4/10/2008
6 – Low	5/2/2008
7 – Don't Fix	7/14/2008



Buffer in the schedule

- New feature ideas
- Responding to the competition
- Integration (getting everyone's code to work together when it's merged)
- Debugging time
- Usability testing (and incorporating the results of those tests into the product).
- Beta tests

Will it converge and lower?



A few more points:

1. Only the programmer doing the work can create the estimate.
2. Fix bugs as you find them, and charge the time back to the original task.
3. Don't let managers badger developers into shorter estimates.
4. A schedule is a box of wood blocks.

"use it as a chance to remove unnecessary features"

Can we use this for our daily tasks?