Review

- QA
- Risk management
- Life-Cycle Models
- Estimation
  - size → effort → schedule
- Size estimation
  - Function points

1

Approximate Language Levels

Language	Level	Statement per Function Point
Excel, Lotus 123, Quattro Pro, etc.	~50	6
Smalltalk 80; Smalltalk/V	15	20
AWK	15	25
Perl	15	25
SAS, SPSS, other statistics packages	10	30
Visual Basic 3	10	30
dBase IV	9	35
Paradox	9	35
Focus	8	40
Oracle	8	40
Sybase	8	40
C++	6.5	50
Quick Basic 3	5.5	60
Lisp	5	65
Ada 83	4.5	70
Modula 2	4	80
Cobol (ANSI 85)	3 - 5	90
Pascal	3.5	90
GW Basic	3.25	100
Fortran 77	3	110
C	2.5	125
Macro assembler	1.5	215
Assembler	1	320

*Adapted from RD

2

Lecture 4

Scheduling
Virtualization
UML - Introduction

3

Project Schedule Estimation

4

Schedule Estimation

- The software is divided into three categories:
  - System: the system software defined here excludes firmware, real-time embedded systems, avionics, process control, etc.
  - Business: business applications such as accounting, reporting, …
  - Shrink-wrap: packaged software, such as Microsoft Office, Adobe creative suite, …
- Three ballpark estimation methods
  1. Function point → effort
  2. Schedule-effort equation → schedule
  3. Schedule tables: LOC → effort, schedule

5

Schedule Estimation (1/3)
Jones's First-Order Estimation

$$Schedule(months) = [Function\ Point]^{power}$$

Kind of Software	Best in Class	Average	Worst in Class
Systems	0.43	0.45	0.48
Business	0.41	0.43	0.46
Shrink-wrap	0.39	0.42	0.45

- Based on the appropriate combination of the kind of software and the organization's class, an exponent can be determined from the above table
- This is an empirical method based on several thousand projects.
- E.g.
  - Developing a business software with an average organization → 0.43
  - $350^{0.43} \sim 12$ calendar months

6

Schedule Estimation (2/3)
Schedule-Effort Equation

$$Optimal\ Schedule = 3 \times Effort^{1/3}$$

$Optimal\ Schedule : months$
$Effort : man-months$

Assuming an effort estimation gives the effort of 65 man-months
Optimal schedule: 12 months ($3.0 * 65^{1/3}$)
Optimal team size: 5 – 6 team members (65/12)

7

Schedule Estimation (3/3)
Schedule Tables

- Shortest possible schedule. (impossible to beat)
  - Talent from top 10%, years of experience in environment
  - Ideal management, all staff available day 1
  - Requirements known day 1 and don't change
  - Tools, offices, methods are ideal
- Efficient schedule.
  - Talent from top 25%, low turnover
  - Competent management, staff available as needed
  - Requirements changes are minor (5%); tools, offices are effective
- Nominal schedule
  - Talent from top 50%, turnover 12% per year
  - Some familiarity with tools and environment

8

Shortest Possible Schedule Table

	System Products		Business Product		Shrink-Wrap Product	
System Size (lines of code)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)
10,000	6	25	4	5	4	8
15,000	7	40	4	8	5	13
20,000	8	57	5	11	6	19
25,000	9	74	5	15	6	24
30,000	9	110	6	22	7	37
35,000	10	130	6	26	7	44
40,000	11	170	6	34	7	57
45,000	11	195	6	39	8	66
50,000	11	230	7	46	8	79
60,000	12	285	7	57	9	98
70,000	13	350	8	71	9	120
80,000	14	410	8	83	10	140
90,000	14	480	9	96	10	170
100,000	15	540	9	110	11	190
120,000	16	680	10	140	11	240
140,000	17	820	10	160	12	280
160,000	18	960	10	190	13	335
180,000	19	1,100	11	220	13	390
200,000	20	1,250	11	250	14	440
250,000	22	1,650	13	330	15	580
300,000	24	2,100	14	420	16	725
400,000	27	2,900	15	590	19	1,000
500,000	30	3,900	17	780	20	1,400

Source: RD

9

Efficient Schedule Table

	System Products		Business Product		Shrink-Wrap Product	
System Size (lines of code)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)
10,000	8	24	5	5	6	8
15,000	10	38	6	8	7	12
20,000	11	54	7	11	8	18
25,000	12	70	7	14	9	23
30,000	13	97	8	20	9	32
35,000	14	120	8	24	10	39
40,000	15	140	9	30	10	49
45,000	16	170	9	34	11	57
50,000	16	190	10	40	11	67
60,000	18	240	10	49	12	83
70,000	19	290	11	61	13	100
80,000	20	345	12	71	14	120
90,000	21	400	12	82	15	140
100,000	22	450	13	93	15	160
120,000	23	560	14	115	16	195
140,000	25	670	15	140	17	235
160,000	26	709	15	160	18	280
180,000	28	910	16	190	19	320
200,000	29	1,300	17	210	20	360
250,000	32	1,300	19	280	22	470
300,000	34	1,650	20	345	24	590
400,000	38	2,350	22	490	27	830
500,000	42	3,100	25	640	29	1,100

Source: RD

10

Nominal Schedule Table

	System Products		Business Product		Shrink-Wrap Product	
System Size (lines of code)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)
10,000	10	48	6	9	7	15
15,000	12	76	7	15	8	24
20,000	14	110	8	21	9	34
25,000	15	140	9	27	10	44
30,000	16	185	9	37	11	59
35,000	17	220	10	44	12	71
40,000	18	270	10	54	13	88
45,000	19	310	11	61	13	100
50,000	20	360	11	71	14	115
60,000	21	440	12	88	15	145
70,000	23	540	13	105	16	175
80,000	24	630	14	125	17	210
90,000	25	730	15	140	17	240
100,000	26	820	15	160	18	270
120,000	28	1,000	16	200	20	335
140,000	30	1,200	17	240	21	400
160,000	32	1,400	18	280	22	470
180,000	34	1,600	19	330	23	540
200,000	35	1,900	20	370	24	610
250,000	38	2,400	22	480	26	800
300,000	41	3,000	24	600	29	1,000
400,000	47	4,200	27	840	32	1,400
500,000	51	5,500	29	1,100	35	1,800

Source: RD

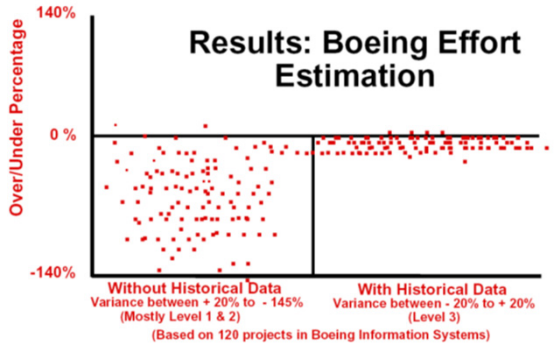
11

Schedule Estimates

- By using function points or lines of code, we can now get a ballpark estimate (better than nothing) on the schedule of the software from the previous two empirical methods.
  - Count FP → Schedule (months) = FP^{power} → # Months = 3 * $Effort^{1/3}$ → # people = Effort / Months
  - Count FP → p. **2** converts FP to LOC → Use on of the schedule tables on p. **9, 10 or 11** → # of people, # months
- The schedule estimates from these empirical methods exclude the time for requirement specification!
- The best schedule estimate, however, needs to be based on **historic performance**!

12

Importance of Historical Data



13

Source: CMMI V1.1 and Appraisal Tutorial, Mike Phillips (2004)

Schedule compression equations

$$\text{Schedule Compression Factor} = \frac{\text{Desired Schedule}}{\text{Initial Schedule}}$$

$$\text{Compressed schedule effort} = \frac{\text{Initial effort}}{\text{Schedule compression factor}}$$

Initial schedule estimate: 12
Initial effort estimate: 78 man-months

Want compress schedule to 10 → SCF = 10/12 = 0.83
Compressed schedule effort: 78/SCF = 94 (man-months)

Some researches conclude that it is impossible to have SCF < 0.75 or 0.80
These equations can also be used to "uncompress" a schedule

14

Review: Estimation and Scheduling

- LOC: Lines of Code
- FP: Function Points
- Counting FP
  - FP $^{(0.43)}$ → calendar months → Schedule-Effort Equation → effort → team size
  - FP → LOC → Schedule Tables → effort, calendar months
- **Applying estimation-convergence graph yields a ranged-estimation**

15

Summary

Project Estimation: size, effort, schedule
size: LOC, FP
effort: man-months
schedule: months

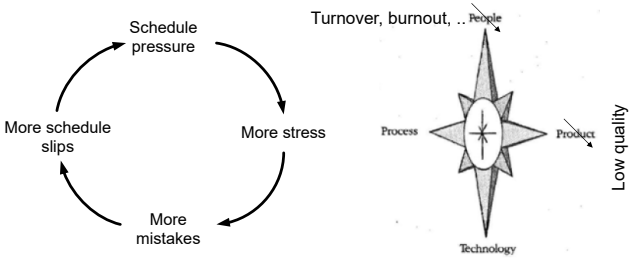
Schedule Pressure & Feature Set Control

16

Scheduling

Scheduling Problems

- Developers underestimate their projects by 20 – 30%.
- Average small-project estimate is off by > 100%.



Silver bullet syndrome

40% of software errors are due to schedule pressure

WinWord 1.0

- Microsoft Word for Windows 1.0
  - Example of overly optimistic schedule
  - 5 years in development, 660 man-months, 249,000 lines of code
    - check nominal schedule tables → 26 months, 800 man-months
  - The final 5-year schedule was approximately five times as long as originally planned.

Report Date	Estimated Ship Date	Estimated Days to Ship	Actual Days to Ship	Relative Error
Sep-84	Sep-85	365	1887*	81%
Jun-85	Jul-86	395	1614	76%
Jan-86	Nov-86	304	1400	78%
Jun-86	May-87	334	1245	73%
Jan-87	Dec-87	334	1035	68%
Jun-87	Feb-88	245	884	72%
Jan-88	Jun-88	152	670	77%
Jun-88	Oct-88	122	518	76%
Aug-88	Jan-89	153	457	67%
Oct-88	Feb-89	123	396	69%
Jan-89	May-89	120	304	61%
Jun-89	Sep-89	92	153	40%
Jul-89	Oct-89	92	123	25%
Aug-89	Nov-89	92	92	0%
Nov-89	Nov-89	0	0	0%

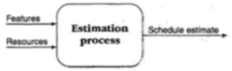
Source: RD

Schedule Pressure

- Cause
  - wishful thinking by customers, managers, …
  - Little awareness of the software estimation methods
  - Poor negotiating skills
    - 75% developers are introverts (where only 33% of general population are)
    - Managers tend to be 10 years older and negotiate for a living
    - Developers oppose negotiating tricks (e.g. high initial estimates)
- Resolution
  - Principled Negotiation
    - Separate people from the problem (cooperate, explore options)
    - Focus on interests, not positions (find underlying needs)
    - Find options for mutual gain (phasing, fewer features, add resources)
    - Insist on using objective criteria (don't negotiate the estimate itself)

Feature set reduction!

Increase resource allocation



Feature Set Control

- Early project: feature set reduction
  - Minimal specification
  - Requirement scrubbing
  - Versioned development
- Mid-project: feature creep control
  - Setup a change-control board (to review/accept/reject changes)
  - Versioned development
  - Sort development cycles
- Late project: feature cuts
  - Eliminate low priority features
- Keep in mind:
  - 50% cut in project size yields a 75% reduction in resources and ~50% reduction in schedule

Phases in Waterfall Model

- Software concept
- Requirement analysis
- Life-cycle planning & technology selection
- Architectural Design
- Detailed Design
- Coding & Testing
- System testing

Assignment #2

Due: 10/19/2015

Based on your business proposal from assignment #1, document the system requirements (functions and features you need). The document to be handed in should consist of the following:

- Write a short description of purposes of your software products.
- List all functions of each product and give priorities for each function.
  - It should be noted that you should consider functions from different roles: regular customers, your own worker / administrator, VIP, etc. …
  - You should also give clear definition of each role, and there should be no ambiguity for each function that you listed. (e.g. How to be a VIP.)
  - You may use UML use-case diagram (to be covered) to document these functions.

Assignment #2

- For each function, identify its input and output.
- Identify core functions (minimal feature set) that is necessary to maintain your business operations.
  - Now you should have a total of four products in your planning:
    - A two-tier desktop application with all functions
    - A three-tier web application with all functions
    - A two-tier desktop application with core functions
    - A three-tier web application with core functions
- Analyze **function points** for all four products.
- Based on the introduced life-cycle models, identify which life-cycle model should you use for your project. State your reasons concisely.
- Estimate effort and schedule for each of these products.