

1.264 Lecture 4

Time and resource estimation, part 2

Next class: Read CMMI papers. Hand in case studies by noon before class

Schedule estimation from tables

- **Shortest possible schedule. (You can't beat this.)**
 - 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

Schedule estimation from tables

Shortest Possible Schedules						
	<i>Systems Products</i>		<i>Business Products</i>		<i>Shrink-Wrap Products</i>	
System Size (lines of code)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)
10,000	6	25	3.5	5	4.2	8
15,000	7	40	4.1	8	4.9	13
20,000	8	57	4.6	11	5.6	19
25,000	9	74	5.1	15	6	24
30,000	9	110	5.5	22	7	37
35,000	10	130	5.8	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

Image by MIT OpenCourseWare.

Schedule estimation from tables

Efficient Schedules						
	<i>Systems Products</i>		<i>Business Products</i>		<i>Shrink-Wrap Products</i>	
System Size (lines of code)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)	Schedule (months)	Effort (man-months)
10,000	8	24	4.9	5	5.9	8
15,000	10	38	5.8	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

Image by MIT OpenCourseWare.

Schedule estimation from tables

Nominal Schedules						
	<i>Systems Products</i>		<i>Business Products</i>		<i>Shrink-Wrap Products</i>	
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

Image by MIT OpenCourseWare.

Questions

- How long would it take to write a 30,000 line systems product with the three different approaches (fastest, efficient, nominal)?
 - How large would the team be in each case?
 - Explain the differences
- How long would it take to write 50,000 line systems, business and shrink-wrap products with a nominal approach?
 - How large would the team be in each case?
 - Explain the differences
- Graph calendar months, person months versus lines of code for a systems product, any approach
 - Describe whether it's linear or nonlinear
 - Just graph a few points; you don't need all of them
 - If nonlinear, in what way? Economies or diseconomies of scale?
- You don't need to apply ranges in this exercise

Answers

- **30,000 line systems product with the three different approaches?**
 - 9 months fastest possible
 - 13 months efficient
 - 16 months nominal
- **How large would the team be in each case?**
 - 12 people fastest possible
 - 7-8 people efficient
 - 11-12 people nominal
- **Explain the differences**
 - Staff quality, process maturity (people, process)
- **50,000 line product with nominal approach**
 - 20 months system, 11 months business, 14 months shrink-wrap
- **Team size**
 - 18 system, 6.5 business, 8 shrink-wrap
- **Explain the differences**
 - Systems software requires much more design, care in implementation, testing, and will have many more bugs
 - Shrink wrap software must be more general and more easily configured and supported than business software
- **Graphs: Nonlinear. Resources (person months) have diseconomies of scale. Schedule time growth less than linear. Team size grows very quickly.**

Estimate refinement

Example of Single-Point-Estimation History

Point in Project	Estimate (man- months)
Initial product concept	100
Approved product concept	100
Requirements specification	135
Product design specification	145
Detailed design specification	160
Final	170

Image by MIT OpenCourseWare.

Example of a Range-Estimation History

Point in Project	Estimate (man- months)
Initial product concept	25-400
Approved product concept	50-200
Requirements specification	90-200
Product design specification	120-180
Detailed design specification	145-180
Final	170

Image by MIT OpenCourseWare.

Scheduling problems

- **Developers/analysts/consultants underestimate task durations by 20-30% on average**
 - And omit 30-50% of tasks
- **Average small project estimate is off by 100%**
 - Big projects are worse
- **Once deadlines are missed, more effort is spent explaining and re-planning**
- **Schedule pressures affect morale, quality**
 - 40% of software errors are due to schedule pressure
 - Gambling in technical approach often occurs
 - This occurs in non-software projects as well

Scheduling pressures

- **Causes**

- **Wishful thinking by customers, managers**
- **No awareness of software estimation methods**
- **Poor negotiating skills**
 - **75% of developers are introverts, only 33% of population is**
 - **Marketers, managers tend to be 10 years older and negotiate for a living**
 - **Developers oppose negotiating tricks (high initial estimates, etc.)**

- **Cures**

- **Principled negotiation**
 - **Separate people from positions (cooperate, explore options)**
 - **Focus on interests, not positions (find underlying needs)**
 - **Find mutual gains (phasing, fewer features, add resources)**
 - **Insist on using objective criteria (don't negotiate the estimate itself)**

Scheduling: Feature set control

- **Early project: feature set reduction**
 - Minimal spec
 - Requirements scrubbing
 - Versioned development
- **Mid-project: feature creep control**
 - Change analysis, change control board
 - Versioned development
 - Short development cycles
- **Late project: feature cuts**
 - Eliminate low priority features
- **Remember:**
 - A 50% cut in project size yields a 75% reduction in resources and about a 50% reduction in schedule

Scheduling: Recovery

- **Most projects are in recovery mode much of the time**
 - Primary problem is not how to finish quickly, but how to finish at all
- **Options**
 - Cut software/project size
 - Increase productivity with short-term improvements
 - Slip the schedule
- **Recovery plan**
 - People: improve morale, correct major personnel and major leadership problems
 - Adding people to a late project only makes it later
 - Process: fix classic errors, miniature milestones, risk mgt
 - Product: stabilize requirements, cut features, fix bugs

Summary: resource estimation

- **Almost always use spiral/agile model: Multiples of:**
 - Requirements, design, implementation, QA
- **Use requirements and design documents to estimate resources using function points**
 - Requirements define product scope
 - Choose technology, determine lines of code
- **Estimate schedule, personnel**
 - System type (systems, business, shrink wrap)
 - Process type (fastest, efficient, nominal, other)
 - Apply convergence graph to all estimates
- **Adjust as things change**
 - Spirals, negotiation, feature management
 - Mini milestones, risk mgt, avoid classic mistakes
- **This works in non-software projects just as well**
 - Shortest possible schedule: more expensive
 - Bigger project than past: diseconomies, more expensive
 - Less talented team than usual: longer, more expensive

MIT OpenCourseWare
<http://ocw.mit.edu>

1.264J / ESD.264J Database, Internet, and Systems Integration Technologies
Fall 2013

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.