Scheduling and Planning

Marko Schütz-Schmuck

Department of Mathematical Sciences
University of Puerto Rico at Mayagüez
Mayagüez, PR

August 13, 2018

Management and Planning

- tracking data
  - all processes discussed here need actual data
  - the engineer needs to keep a time log on all activities
- tasks are categorized
- differences among processes concerning e.g. accuracy of time tracking

Processes

- Personal Software Process
- Team Software Process
- Evidence Based Scheduling
- Managing the Process

Personal Software Process - Introduction

- taught inside and outside universities
- data shows its effectiveness in improving engineers' planning performance and quality of their product
- relies mainly on personally collected data
- defines software process measures for analyzing your data for some different quality goals
- software process is then modified
- measurements of changed process evidence whether goal is reached

Observations on Time Management

- you will likely spend your time this week much the way you spent time last week.
- to make realistic plans, you have to track the way you spend time.
- to check the accuracy of your time estimates and plans, you must document them and later compare them with what you actually do.
- to make more accurate plans, determine where your previous plans were in error and what you could have done better.
- to manage your time, plan your time and then follow the plan.

Understanding Your Time Spending

- categorize your major activities.
- record the time spent on each major activity.
- record time in a standard way.
- keep the time data in a convenient place.

Keeping Notes

- time estimates and actual time spent
- also
  - assignments
  - design ideas
  - insights
- purpose
  - data gathering
  - evidence to support diligence in case something goes wrong
  - evidence to establish invention date in case of patent applications

Tracking Your Time

- usually spend less than one hour on task
- use minutes to record time spent

Time Tracking Format

Date	activity happened on
Start	obvious
Stop	obvious
Interruption	any time lost due to interruptions
Delta time	net time spent on the activity, in minutes
Activity	descriptive name
Comments	notes/comments helpful when analyzing the time data
Completed	task was completed
Units	task size consisted of this many units

Interruptions and other Events

- PSP requires carefully recording interruptions
  - goal: learn to accurately predict pure work time for tasks
  - secondary effect: become more aware and sensitive to interruptions, can better control number and length of interruptions
- other approach to interruptions: EBS (later)
- mark completed tasks
- record the units the task was long: e.g. number of pages read, lines of source code written

Tracking Completeness and Accuracy

- want complete and accurate data
- keep time log with you at all times (e.g. note-taking app on phone)
- in case you forget: track time on napkin or whatever else and update master log asap
- in case you forget tracking an activity: estimate and update master log asap
- use stopwatch to track interruptions (seems excessive, but is quite simple)
- summarize promptly (e.g. weekly)

Period and Task Plans

- complementary perspectives
  - period plan: how do you plan to spend time during a given calendar period (day, week, . . .)
  - task plan: how much time you plan to spend on a task
- example:
  - product plan: reading all 20 chapters of some textbook in 20 hours
  - period plan: spend 1 hour every workday on reading
  - combined: read entire textbook over coming 4 weeks

Time Recording Log

Name: Angel Velez Date: 8/July/18

date	start	stop	interrupt	net	act	comment	C	U
9/Jul	9:30	10:20		50	class	lecture		
	12:50	1:18		38	prog	HW 1		
	2:45	3:53	10	58	prog	HW 1		
	6:25	7:45		80	text	read ch 1 & 2	x	2
10/Jul	11:06	12:19	6+5	62	prog	HW 1, break, chat	x	1
11/Jul	9:30	10:20		50	class	lecture		
	1:15	2:35	3+8	69	prog	HW 2, break, phone	x	1
	4:18	5:11	25	28	text	read ch 3, chat mary	x	1
12/Jul	6:42	9:04	10+6+12	114	prog	HW 3	x	1
13/Jul	9:30	10:20		50	class	lecture		
	12:38	1:16		38	text	read ch 4		
14/Jul	9:15	11:59	5+3+22	134	rev	quiz prep, br, ph, chat		

Weekly Activity Summary I

Name: Angel Velez Date: 14/July/18

week#	Task	Class	Write	Quiz	Read	Total
1	Date		Prog	Prep	Text	
1	S 8/Jul					0
2	M	50	96		80	226
3	T		62			62
4	W	50	69		28	147
5	T		114			114
6	F	50			38	88
7	S			134		134
8	Totals	150	341	134	146	771
	Previous					0
	Week's					0
9	total					0
10	avg					0
11	max					0
12	min					0
	Current					0
	Week's					0
13	total	150	341	134	146	771
14	avg	150	341	134	146	771
15	max	150	341	134	146	771
16	min	150	341	134	146	771

Weekly Activity Summary II

Name: Angel Velez Date: 21/July/18

week#	Task	Class	Write	Quiz	Read	Total
2	Date		Prog	Prep	Text	
1	S 15/Jul					0
2	M	50	93		80	223
3	T		95			95
4	W	50			71	121
5	T		77			77
6	F	50	74		40	164
7	S				33	33
8	Totals	150	339	0	224	713
	Previous					
	Week's					
9	total	150	341	134	146	771
10	avg	150	341	134	146	771
11	max	150	341	134	146	771
12	min	150	341	134	146	771
	Current					
	Week's					
13	total	300	680	134	370	1484
14	avg	150	340	67	185	742
15	max	150	341	134	224	771
16	min	150	339	134	146	713

Making Task Plans

- plan all projects or major tasks
- first step: clear definition of the task, project, or product
- know "what" before you consider "how"
- task plan includes
  - size and important features of major task, project, or product
  - estimate of time required
  - projection of schedule
- more complex projects require additional information: staffing plans, dependencies on other groups, etc
- from now on: make a plan for every major task
- if you estimate it takes days rather than hours: break it down

Job Number Log I

Name: Angel Velez Date: 21/July/18

job#	description	date	process	est	est	act	act	act	agg	agg	agg	agg	agg
				time	units	time	units	rate	time	units	rate	max	min
1	write prog 1	9/Jul	prog	100	1	158	1	158.0	158	1	158.0	158	158
2	read ch 1&2	9/Jul	read	50	2	80	2	40.0	80	2	40.0	40	40
3	write prog 2	11/Jul	prog	158	1	69	1	69.0	227	2	113.5	158	69
4	read ch 3	11/Jul	read	40	1	28	1	28.0	108	3	36.0	40	28
5	write prog 3	12/Jul	prog	114	1	114	1	114.0	341	3	113.7	158	69
6	read ch 4	13/Jul	read	60	1	118	1	118.0	226	4	56.5	118	28
7	write prog 4	16/Jul	prog	114	1	93	1	93.0	434	4	108.5	158	69
8	write prog 5	17/Jul	prog	109	1	95	1	95.0	529	5	105.8	158	69
9	read ch 5	18/Jul	read	57	1	71	1	71.0	297	5	59.4	118	28
10	write prog 6	19/Jul	prog	106	1	151	1	151.0	680	6	113.3	158	69
11	read ch 6	20/Jul	read	59	1	40	1	40.0	337	6	56.2	118	28
12	read ch 7	21/Jul	read	56	1								

Job Number Log II

Name: Angel Velez Date: 8/July/18

date	start	stop	interrupt	net	job	comment	C	U
9/Jul	9:30	10:20		50	class	lecture		
	12:50	1:18		38	1	HW 1		
	2:45	3:53	10	58	1	HW 1		
	6:25	7:45		80	2	read ch 1 & 2	x	2
10/Jul	11:06	12:19	6+5	62	1	HW 1, break, chat	x	1
11/Jul	9:30	10:20		50	class	lecture		
	1:15	2:35	3+8	69	3	HW 2, break, phone	x	1
	4:18	5:11	25	28	4	read ch 3, chat mary	x	1
12/Jul	6:42	9:04	10+6+12	114	5	HW 3	x	1
13/Jul	9:30	10:20		50	class	lecture		
	12:38	1:16		38	6	read ch 4		
14/Jul	9:15	11:59	5+3+22	134	rev	quiz prep, br, ph, chat		
16/Jul	9:30	10:20		50	class	lecture		
	2:10	4:06	4+19	93	7	HW 4, br, ph	x	1
	7:18	8:49	11	80	6	read ch 4, chat	x	1
17/Jul	9:26	11:27	4+22	95	8	HW 5, br, ph	x	1
18/Jul	9:30	10:20		50	class	lecture		
	4:21	5:43	11	71	9	read ch 5, br	x	1
19/Jul	6:51	9:21	51+16+6	77	10	HW 6		
20/Jul	9:30	10:20		50	class	lecture		
	12:33	1:18	5	40	11	read ch 6, br	x	1
	1:24	2:38		74	10	HW 6	x	1
21/Jul	11:18	11:51		33	12	read ch 7	x	1

Using Data for Planning

- after 6 programs written: average 113.3 minutes
- simple planning heuristic: next program estimated ~2hrs

Size Measurement

- process: read textbook
- measure: chapter
- but: different chapters tend to have different number of pages
- better measure: #pages

Cautions on Size Measurements

- introduction/overview/summary chapters are often easier (faster) to read than the core technical chapters
- in general: some texts are much harder to read than others
- prior exposure
    - rereading text you read before, having practiced what's explained
    - reading for the first time
- intent
    - rereading a draft you have written for the first time
    - skimming a draft you have written for e.g. the 10th time

Size Measurements of Programs

- lines of code
- function points
- use case points
- story points

Lines of Code

- easily inflated
- low-level vs high-level programming language
- complexity of source code

Function Points

- categories: inputs, outputs, inquiries, internal files, external interfaces
- shown to correlate with lines of code

Story Points

- user stories
- planning poker

Making Estimates for Larger Programs

- decompose
  - keep data for different kinds of functionality
    - calculation, storage, user interface
  - how large will each of these be? LOC or other measure
  - min, max, avg

Managing Your Time

- use historical data to see how you have spent time in the past
- you will likely spend time in a similar way, but you can make adjustments
- some categories are obvious from the start, others occur seldom: try to foresee these also
- balance categories
  - category taking >=50% of your time: decompose?
  - category taking <=3% of your time: consolidate?

Evaluating Time Spending

- which tasks are most important?
- do I spend enough time on important tasks?
- do some tasks take too long, keeping me from other important tasks?
- am I allowing enough time for e.g. homework?
- what are my personal commitments?
- these are very personal and sometimes complex decisions
- do not burn yourself out!
  - allow for job, studies, sleep, physical activity, social and family life

Using a Time Budget

- use historical data to see how you have spent time in the past
- plan how much time you will spend on each category of activities
- key to time management: gradually re-balance
  - adding 20 minutes more per week to one activity category might be realistic
  - adding 3 hours more per week to one activity category might be unrealistic
- goal: sustainable time budget

Finding more Time

- you will not
- you can reallocate
  - previously unallocated time can be allocated: rare
  - consider all your commitments
    - you may have to compromise
    - e.g. each week you do 6 physical exercise sessions at 75min + 15min (shower etc.), total 540min
    - switch to 3 sessions 100min + 15min (shower etc) and 3 mobility, light stretching (no-sweat) sessions 30min + 5min (changing), total 450min
- you need a time budget that you *want* to follow

Overall Weekly Time Summary

activity	CS	English	Spanish	Math	eat/rest	other	total
fixed							
class	150	150	150	150			600
required							
HW	360	240	240	360			1200
read	240	60	90	150			540
discretionary							
eat					1260		1260
sleep					3360		3360
phys. act.						450	450
entertainment						840	840
relaxation						1830	1830
total	750	450	480	660	4620	3120	10080

Weekly Time Schedule

#=latex:

task	class	write	exam	read	rev	other	total
		prog	prep				
day							
S							
M	9:30-10:20 10:30-11:20 1:30-2:20			3:00-4:00			210
T	12:00-1:15	8:30-10:30		3:00-4:00			255
W	9:30-10:20 10:30-11:20 1:30-2:20			3:00-4:00			210
T	12:00-1:15	8:30-10:30		3:00-4:00			255
F	9:30-10:20 10:30-11:20 1:30-2:20			3:00-4:00			210
S		8:30-10:30		3:00-4:00			180
total	600	360		360			

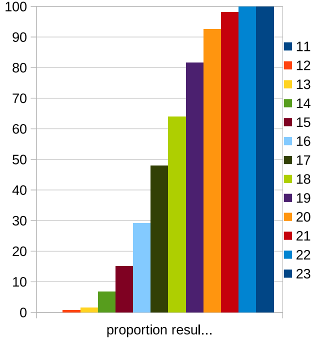
Thought on Variable Time

- what are your highest priority items?
- are there some tasks that should be done at specific times?
- are there activities you want to do as soon as you have the time?

careful combining averages: Monte Carlo Simulation I

estimates (days)	best-case	most likely	worst-case
tasks			
writing content	4	6	8
creating graphics	5	7	9
multimedia integration	2	4	6
total duration	11	17	23

careful combining averages: Monte Carlo Simulation II



- only 50% of the simulations complete on or before 17 days
- almost 93% of the simulations complete on or before 20 days

TaskJuggler

- `http://taskjuggler.org`
- text-based description of project
- for larger projects

Work Breakdown Structure

- deliverable-oriented
- itself a deliverable
- starting with overall project deliverable decompose into manageable components
  - size
  - duration
  - responsibility
- tree structure

WBS Principles

100% rule captures all deliverables

no overlap components are mutually exclusive

outcomes do not prescribe actions or methods, but outcomes or results

detail terminal elements can be realistically estimated, make no sense to further decompose, produce a measurable deliverable

80 hour rule terminal elements to produce one deliverable <= 80 hour effort and not longer than reporting period