The Renzo Pomodoro Dataset: a conversation

Derek M. Jones
Knowledge Software
derek@knosof.co.uk

Renzo Borgatti
Droit LLC
renzo.borgatti@droit.tech

December 12, 2019

Abstract

An individual's estimated Pomodoros for daily planned tasks, over a period of 10 years, is analyzed (14,503 planned tasks, with 10,555 unique task descriptions).

Findings todate include: Tasks estimated to take 1, 2 and even numbers of Pomodoro are in good agreement with the actual Pomodoro; various reasons for why estimates of 3, 5 and 7 Pomodoro are in poor agreement with actual are discussed...

This is a work in progress...

1 Introduction

This paper is a work in progress...

This paper takes the form of a conversation between Renzo Borgatti, who kept a record of tasks he planned and completed, over a period of 10-years, and Derek Jones who analysed the data. Renzo used the Pomodoro Technique, a time management technique developed by Francesco Cirillo.[1] Renzo applied the Technique to the best of his knowledge, sometimes adapting its features to his personal taste. Although the paper presents data about the Technique, it's not going into the details of how to use it, nor trying to judge the Technique in general.

Data analysis is an iterative process; ideas may have been suggested by discussions with those involved before the data arrives, and new ideas are suggested by feedback from the ongoing analysis. Most ideas go nowhere; failure of the data to support an idea is the norm. Analysts who are not failing on a regular basis, never discover anything.

The reason for doing data analysis is to obtain information that is useful to those involved with the system that produced the data.

Any collection of measurements contains patterns, and some of these may be detected by the statistical techniques used. Connecting patterns found by data analysis to processes operating in the world requires understanding something about the environment and processes that generated the data. If the person doing the data analysis is not intimately familiar with this environment and processes, they either have to limit themselves to generalities or work as a part of a team, with people who have this knowledge.

As this conversation progresses, the narratives created as possible explanations for the patterns found in the data evolve; readers are not presented with a well-structured story fitted together after the event.

The data, along with the R scripts used for the analysis, are available at: `https://github.com/Derek-Jones/renzo-pomodoro`

1.1 Stumbling onto data

Derek

I first met Renzo after talking at a meetup he organized[1]. After the talk, Renzo told me that for many years he had been recording his daily work tasks, along with estimated and actual Pomodoro counts.

Renzo was describing a kind of dataset that I had not previously encountered, and I was not sure what to make of it; however, I was willing to have a look at it, to try to find some useful patterns. Renzo promised to send me a copy of the data.

My professional background is compiler writing and source code analysis. Over the last seven years I have collected and analyzed 500+ software engineering datasets and made them publicly available.[3]

Renzo

I started using the Pomodoro Technique sometimes around 2005/2006. At the beginning I used pencil, paper and a timer (a Pomodoro-shaped kitchen timer exactly like the one that gives the name to the Technique) but probably because of my work as software developer, I immediately found easier to record Pomodoros as text files on my computer. My Pomodoro records go back to 2006. However, it's only after 2009 that I started recording them on the same file format that I still use today.

My daily cycle goes as follows: I start a 25 minutes timer and copy yesterday's Pomodoro plan into today's plan (e new text file with a timestamped name). Then I move the worked Pomodoros from the day before into the "THIS WEEK" section of the file and change the top date to today's date. Then I start planning what I want to do today, writing a descriptive line for each task I want to accomplish. The lines look like the following:

```
@planning proper 1
@meeting company updates 2
@project @x encode new rules into stylesheets 5
@paper @pomodoro write high level description of the Technique 4
@personal call insurance for claims 1
```

You can see a few tags, which describe the type of task. The number at the end of the line represents the estimated number of Pomodoros I think I'm going to spend. I might add a few more tasks with the same format, until they add up to a total of 10-15 estimated Pomodoros. The tasks appear roughly in

[1] `https://www.meetup.com/Papers-We-Love-London/events/258538644/`

the order I'd like to execute them. At this point, almost every day, I've already spent my first 25 minutes when the timer goes off! At this point I update the first line with:

```
@planning proper 1 => 1
```

Since "@planning" for the day is done, I move the line into "DONE" section. I then append a descriptive log into another file, which contains all Pomodoros worked since the beginning of time, which reads like:

```
@planning for the day and ready to roll!|2019-08-05 09:06
```

This line has a specific format: I repeat the tags from the planned Pomodoro, a free-form description followed by the "|" sign and then a timestamp. I use this new description to describe any insight learned from executing the last Pomodoro. Sometimes this is meaningful, sometimes it's less interesting. Finally, I break a few minutes (perhaps tea, coffee, or select music for the day) and start a new 25 minutes timer.

Hopefully by the end of the day, all tasks planned have moved to the done section. Sometimes it takes longer to finish them, sometimes less. The next day, if there are leftovers from the day before, I consider if to re-schedule them or not based on priorities.

I tried hard to keep the file format consistent over the years, with some success. However, due to the lack of formal tooling built around it, the format evolved over the years and several files might contain errors or inconsistencies, making the task of analyzing the data difficult at times.

1.2 The conversation

Getting the most out of data analysis requires domain knowledge. Renzo has that knowledge, but is a very busy man. I find the best way to get a busy person to talk to me, is to tell them things they find interesting and useful. Derek

My top priority is to find something in the data that the domain expert finds interesting and useful. The boring, but necessary, stuff can be done later.

1.3 The Renzo dataset

The Renzo dataset arrives in raw form, i.e., the files containing text in the structured format used by Renzo. Various `awk` scripts were written to extract the available information on each task, and to generate a corresponding row of values in a csv file. Derek

An initial analysis needs to answer two questions:

- do I believe the data? This question is not about whether the data was fabricated, but whether the information present is likely to be a reasonably accurate representation of what it claims to be. People make mistakes, decimal points may be misplaced, times and dates entered later are misremembered, measurements are made using different units by different people (e.g., miles vs. kilometers),

- what information is present in each column and what properties does it have?

One method for getting a quick overview of data, is to look at it within an editor (although, if there are lots of columns, line wrap can complicate the visualization). R's `str` function lists column names, the base-type of the data and the first few values.

```
'data.frame': 14503 obs. of  7 variables:
 $ X.words    : chr  "@general" "@meta" "@metrics" "@charts" ...
 $ word_cnt   : chr  "1,1,1,1,1,1,1" "1,1,1,1,1,1,1,1,1" "1,1,1,1,1,1,1,1,1,1,1,1,1,1
 $ description: int  3673 8689 9183 9979 9447 1 1049 1165 5281 5801 ...
 $ DONE       : int  1 1 1 1 0 1 1 1 0 1 ...
 $ date       : Date, format: "2009-04-09" "2009-04-09" ...
 $ estimate   : num  1 1 2 1 4 1 1 1 2 2 ...
 $ actual     : num  1 2 1 1 NA NA 1 2 NA 2 ...
```

The extracted data contains one row for each planned and completed task. Rows contain the date, estimated and for completed tasks actual effort (in Pomodoros), "@words" associated with the task, and anonymised task description (number of occurrences of each unique word in the description)...

On 1,827 days between February 2009 and February 2019, tasks intended to be done that day were planned. The data lists 14,503 tasks for this period, however, tasks are not always completed on the day they are planned, the DONE sections list 6,633 tasks.

Plotting is an easy-to-use visualization technique that can help highlight patterns in data, for little effort.

Each task has an associated description; the description contains confidential information, and each description has been replaced by a unique number in the released data (the `word_cnt` column lists the number of occurrences of each unique word in a task's description). The `estimate` and `actual` columns specify values measured in Pomodoros.

If the work on a task is completed, the task is moved to DONE. Figure 1 shows the number of unique tasks descriptions that appeared on a given number of days. The tasks are split into those descriptions that only appear in DONE (i.e., the task was completed on the day it was created; 37.2% of all planned tasks), those that never appear in DONE (26.5%), those that maybe DONE (they never appear in DONE, but are listed as involving an actual number of Pomodoro; 16.4%), and those that appeared outside of DONE and one or more days later appeared in DONE (19.9%).

The plot in Figure 1 reveals a few interesting facts. There are around 5,000 unique tasks that never appear in a "DONE" section in any daily file. The ideal lifecycle of a task is to start execution, becoming "DONE" hopefully within the expected number of uninterrupted Pomodoros. This is the procedure with additional details: Renzo

- The task is planned for a specific day with the usual format: a few "@words" tags to classify it, then a description, then an estimate (for ex-

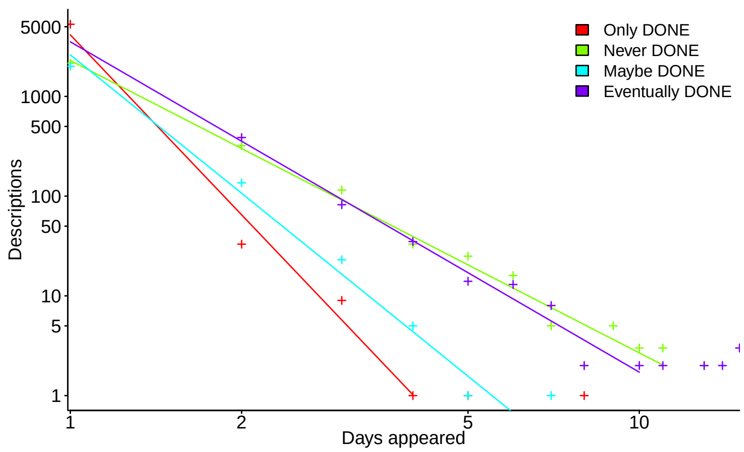

Figure 1: Number of unique task descriptions appearing in a daily task list, on a given number of days.

ample: "@paper write a few words about the lifecycle of a task 4" means I'm planning to work 4-Pomodoros on a task classified as "@paper").

- Each Pomodoro spent on the task increases it's "actual" count. The line remains in place, but it's updated with an arrow "=>" indicating the Pomodoro spent (e.g. "@paper write about the lifecycle of a task 4 => 1" means one Pomodoro was spent on the task).

- When the task is done, it moves from the original position to the "DONE" section. The task could be done ahead of its estimate (over-estimation) or after (under-estimation).

The above describes an ideal execution cycle, but there are scenarios where things don't go as expected:

- The end of the session (usually end of the working day) arrives before the task is actually completed. The goal of the planning activity is to try to avoid this situation, but sometimes this is unavoidable (e.g., other tasks were longer than expected).

- The task is planned for the day but no work is actually spent on it, never collecting a single Pomodoro of execution (the "=>" arrow never appears or, if it does, it shows "=> 0").

- The task is started, a few Pomodoros spent on it and then the task is abandoned without ever finishing it. This is the "down the wrong path" scenario where it is discovered that the task does not bring any value and a new task is created instead. If working in a team, this could also be the scenario related to someone else working on the task and get it finished.

In all scenarios except the last, the task description never enters a "DONE" section, and remains where it is. This is not necessarily bad, but considering the goal of using the technique is to accomplish planned work, they are not an ideal outcome. By looking at the 5,000 tasks that never appeared in a "DONE" section, I can try to understand what tasks don't get completed and why, and possibly remove what appears to be "wasted work". Some of the "wasted" tasks are still useful because they are part of the explorative process of certain projects, but one should pay attention their number doesn't go over a certain threshold.

Continuing the exploration on the plot in Figure 1, we can see that there are around 200 descriptions of tasks that appear in more than one daily planning file. This corresponds to a task that was planned, but couldn't get completed by the end of the day.

Part of each new daily planning session (usually first thing in the morning), is to go through yesterday's planning to see how well it performed. This small retrospective analysis evaluates if estimates were good, what went wrong (interruptions) and what can be done about them. This is also the time when tasks that couldn't get finished the day before, get a chance of rescheduling the current day with a new estimate. This last aspect explains why there are tasks with exactly the same description repeating over several (possibly contiguous days). As we move right on Figure 1 the red line repetitions quickly drop in number, with a few odd data points that could be related to mistakes in tracking.

Repetitions of "Only Done" tasks (the blue line) is less steep. The blue line in Figure 1 shows the happy path scenario in which a task description only appears once in the "DONE" section of a daily plan scenario. The unique task descriptions (those that appears once in a "DONE" section) are about 4,000, a positive fact considering the total of 8,143 tasks when repetitions are included. It means that roughly 4,000 tasks were executed the day they were planned, as it should be.

The long tail of the blue line in Figure 1 can be explained in terms of tasks repeating with the same description. "@planning" tasks for example, tend to appear every day with the same identical description. Similar repeating tasks are "@meeting" tasks (for those meetings that are predictable in nature and have an expected outcome that can be measured).

The left plot in Figure 2 shows estimated task effort against actual task effort, for every task. Estimates and actuals, measured in Pomodoros, are integer values. Each plotted point in Figure 2 has been jittered, that is a small random value has been added, so that points spread out to give some idea of the number of tasks having a given estimate/actual value. It can be difficult to see any patterns in plots of integer valued data. Derek

The equation fitted by the regression model, which explains 38% of the deviance in the data, is: $Actual = 1.25 \times Estimate^{0.67}$; adding information on the number of daily tasks estimated improves this to 39%.

The estimated/actual pattern is seen in other data,[4] i.e., underestimating for low estimates, and overestimating for the higher estimates.

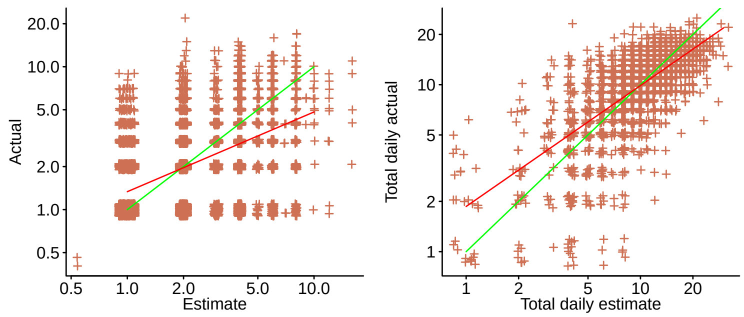

Figure 2: Left: estimated against actual task duration, in Pomodoros, Right: total daily estimated against total daily actual; data has been jittered, green line shows estimated equals actual and red line is a fitted regression model.

Renzo

On one hand it is good to see that estimates are in general consistent. On the other, there are still many under and over-estimates showing up in the charts. In particular, it looks like there are more lower estimates becoming larger pieces of work compared to the opposite. This fact is probably telling me that I should pay attention to tasks that I consider trivial and how they mutate into larger ones.

Derek

The right plot in Figure 2 is based on summing each day's planned tasks; the ideal case (green line), where estimate equals actual, plus a fitted regression model (red line).

The equation fitted by the regression model, explains 79% of the deviance in the data, it is: $Daily\_Actual = 1.4 \times Daily\_Estimate^{0.87}$; adding information on the number of daily tasks estimated improves this to 81%.

What do the @words denote?

Renzo

If a Pomodoro description contains a word starting with "@", that word is interpreted as a "tag". Tags are labels to group similar tasks together. I could for example tag a project, a subproject or a type of activity. The idea is that I can then "search by tag" to filter all Pomodoros spent for some activity I'm interested in. It is essential that tags are used consistently: once tags are selected for a specific task, they should remain the same without variation. Some of them are going to repeat very often (for example, I mark a "@planning" task every day, which is the Pomodoro I dedicate to organize the day). Another frequent tag is "@meeting" as they are in one way or another integral part of my job.

2 Motives and incentives

Derek

What motivates a person to make an estimate, and what are the incentives that influence the thinking process of the person making an estimate?

In a business context there is a framework within which estimation incentives need to operate. For instance is there an existing relationship between supplier and client, and are estimates from multiple suppliers going to be evaluated?

When bidding to be awarded the contract to implement a project, the motivation for investing in making an estimate is the profit to be had from being awarded the contract. The likelihood that others are bidding to be awarded the contract creates a strong incentive to produce the lowest reasonable estimate;[2] once the project is underway the client has little alternative, but to pay more (project overruns receive the media attention, and this is a well-known case).

When an employee is asked to make an estimate for an internal organizational project, the motivation comes from the person who asked for the estimate (i.e., the person's boss, or their boss's boss). As an employee, the person making the estimate may not consider the possibility of competitive bids, and one strategy is to play safe and overestimate; delivering under budget is often seen in a positive light. Underestimates receive little publicity, but are encountered in studies of estimation of organizations' internal tasks.[4]

In many situations it is not cost effective to spend more time estimating, than a rough estimate of the time needed to complete the task. Time spent estimating is likely to be some small percentage of an initial rough guess of the time needed to complete a task.

What motivates an individual to take the time to make these estimates for their private use (I was the second person to see the data, 10-years after the estimation log started), and what are the incentives that might influence them?

I sometimes estimate how long a short task might take me, but don't write anything down. I have worked to a deadline, where what has been completed by a given date/time is what gets shipped (which is not the same as estimating in advance); I have tuned some repetitive tasks, e.g., time to get from home to the train station. I have no insights to offer.

3 An initial analysis

In the first iteration of this data analysis I did not include task estimate data (blue line) in Figure 3 (number of tasks whose estimated/actual effort was a given number of Pomodoros). This was an oversight on my part, and looking at the plot now, shows that it was a major oversight (it appeared to galvanize Renzo). Derek

Estimates and their outcome are an important metric for the technique. After all, the reason why one tries to keep track of how time is spent is to be more efficient. This chart shows that the large majority of the tasks are 1-6 Pomodoros in size. Considering that tasks spans Pomodoros and also breaks (the 5 minutes after each Pomodoro), a task size between 1-6 maximizes the possibility of having at least a few tasks accomplished every day: the probability of finishing a large task of more than 10 Pomodoros is smaller than 5 tasks of 2 Pomodoros each. Renzo

Figure 3 shows the number of tasks whose actual duration required a given number of Pomodoros; the line shows a fitted regression model (its equation is: $Tasks = 5147e^{-0.57Actual}$). Derek

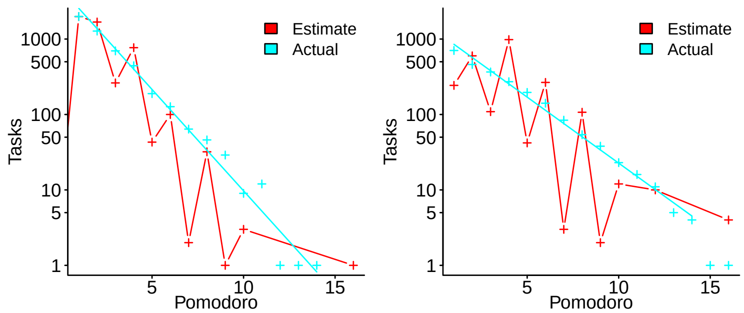

Figure 3: Number of only DONE tasks whose estimated and actual effort was a given number of Pomodoro (left); number of maybe DONE tasks whose estimated and actual effort was a given number of Pomodoro.

Figure 3 clearly shows that odd-valued estimates are significantly under-represented (apart from one Pomodoro). A large improvement in estimation accuracy could be achieved by halving the number of four Pomodoro estimates (I don't have any suggestion for which half), and adding an extra Pomodoro to 40% of the two Pomodoro estimates.

I asked Renzo if he was superstitious.

Renzo

No superstition involved here, although it's an interesting correlation, which I had no idea existed. The preference for even number estimates is possibly a consequence of the fact that the Technique suggests taking a longer break every 4-Pomodoros. I think that subconsciously I might try to fit tasks so the longer break does not happen in between them. However, longer tasks exist in the chart crossing the 4-Pomodoro boundary. For example, how come there is not a single 9-Pomodoro task? Frankly it is hard to say. A contributing factor could be that 9 is so close to 10 that almost all 9-Pomodoro estimates end up being 10-Pomodoros instead. A similar fate happens to 7-Pomodoro tasks (just a few).

Other odd Pomodoro estimates have a strange fate as well, for example 3 and 5-Pomodoro task lengths are consistently underestimated. Size 1 tasks are in large number, probably because there is 1 "@planning" Pomodoro each day that always finishes on time.

Now that I see this habit clearly from the data, I think it shouldn't be the case. I might have a mild interest in not breaking a task across longer breaks of 20-30 minutes, but the design of the working day shouldn't be based on that principle. This chart tells me to think my estimate more carefully, and resist the temptation of just assign them a slot of 4-Pomodoros regardless.

3.1 Data cleaning performed

Derek

Some of the rows in the csv file produced by processing the original text files contain nonsensical task information. Rows matching the following patterns

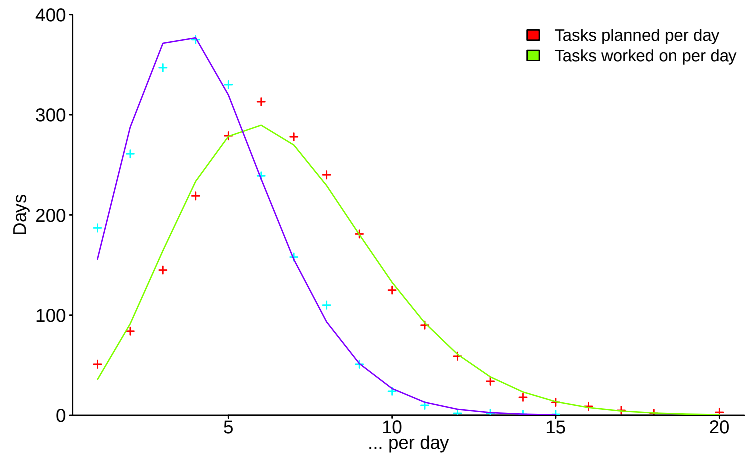

Figure 4: Number of tasks planned per day, and number of tasks worke don per day; lines are fitted zero-truncated type II Negative Binomial distribution.

were removed before performing any of the analysis discussed in this paper (the csv file made available for download contains the original uncleaned data).

- rows without any @word are removed,
- rows with a task estimate greater than 19 are removed,
- on a given day, if the same task description appears both outside of a DONE section and inside a DONE section, the row containing the outside of a DONE section is removed,
- if the same task description appears in a DONE section on two successive days, the row corresponding to the second occurrence is removed,
- spelling correction of @words (see associated R code for details).

Reasons for these nonsensical row values include: the `awk` scripts not correctly handling every pattern found in the original data, and cut-and-paste mistakes made when creating the day's task information.

3.2 Daily totals

Tasks do not occur in isolation, one or more tasks is assembled in to a list of jobs to be done that day. This section analyses daily totals. Derek

Figure 4 shows the number of days on which a given number of tasks were planned, and the number of days on which a given number of tasks were worked on (i.e., have a non-zero actual Pomodoro). Some experimentation finds that a zero-truncated type II Negative Binomial distribution is a reasonable fit to the data (might try some more experimentation with this extra data).

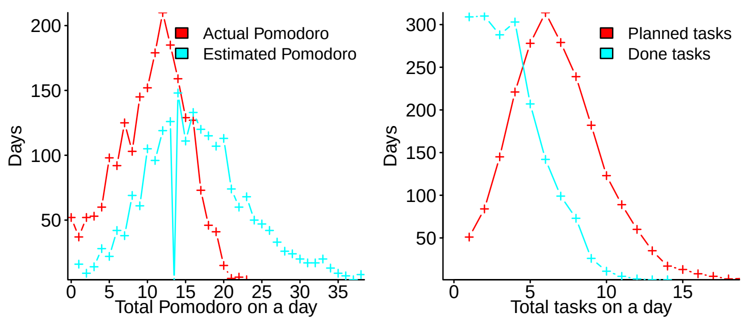

Figure 5: Number of days on which a total number of Pomodoro, estimate and actual, occurred (left), and number of days on which a total number of tasks were planned and done (right).

The left chart on Figure 4 shows the amount of fragmentation of a typical daily session. I call a "session", any amount of contiguous time that is tracked using the PT (typically there is a single session during the day). I'm a bit concerned about all those days with just 1 or 2 planned tasks. Assuming that's correct, then I'm expecting those tasks to be large in size, as a typical daily session averages at 12-13 completed Pomodoros. However, this seems to contradict Figure 3 where large tasks with many Pomodoros are the minority. Renzo

The right plot in Figure 4 shows the number of days on which a given number of Pomodoros were estimated; colored lines show the break-down by the number of tasks performed on a given day. Derek

What is the variation in the daily number of planned tasks and total estimated Pomodoro (for a day)? The left plot in Figure 5 shows the number of days on which a total number of Pomodoro were estimated and actually performed. The right plot shows the number of days on which a given number of tasks were planned, and the number of tasks done.

What is the mean number of actual Pomodoros needed for a task during a day, when one, two, three, etc tasks are estimated on that day?

Figure 6 shows the number of days having a given mean actual task duration, broken down by the number of tasks performed on a day. When more than one task was performed, the mean actual number of Pomodoros roughly centers around two.

Figure 5 gives me an idea of how many days I expected to be productive and uninterrupted. For example I can see that I've estimated 10 Pomodoros (a pretty good estimate for a day) in over 100 days. Higher numbers (such as 12-15 Pomodoros) are still happening on a considerable amount of days. The days with an actual number of Pomodoros (compared to the estimated ones) are consistent with the number of estimated days. Renzo

How are tasks allocated, and how many Pomodoros is a good amount for a given day? I normally used an approach called "Yesterday's Weather": I look at how many Pomodoros I was able to achieve the day before, (sometimes looking

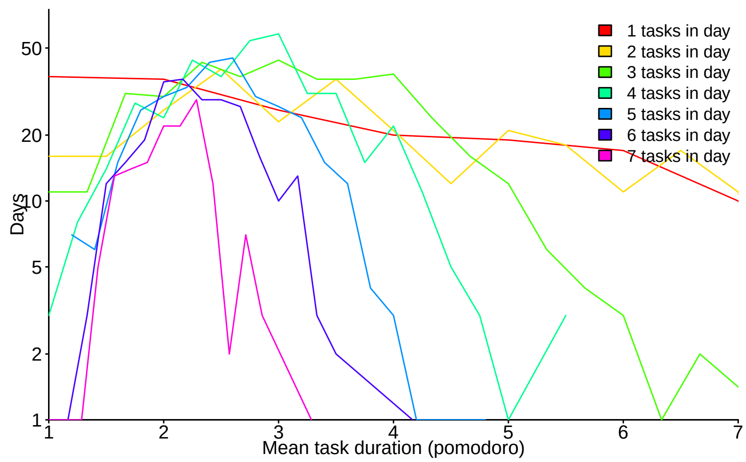

Figure 6: Number of days on which a given mean actual task duration occurred, broken down by number of tasks performed on a day.

at the week before on average as well) to decide how much I should commit for the current day. Factors influencing the decision are the amount of meetings happening throughout the day, or the kind of task.

When I started using the PT, I decided that meetings shouldn't count as Pomodoros. The reason is that meetings are mostly unstructured discussions, so the outcome of a Pomodoro worth of work is difficult to quantify. However, as the years passed, I started taking meeting more seriously, in the sense that I started preparing an agenda and requesting myself or attendees a clear outcome. With that attitude in mind, I decided to start tracking meetings as well. Nowadays I often allocate a "@preparation" tag for a meeting, an "@execution" and a "@conclusion" where I write the meeting minutes. Thanks to the PT (and my need to require an outcome out of every task) I have much more productive meetings today than in the past.

4 @words

The description of every task included one or more @words. The spelling of these @words needed some cleaning to fix obvious typos... Derek

There are 1,056 unique @words, with 271 unique appearances as the first @word, and 890 as the second.

Figure 7 shows the dates on which each @word appears as a task's first @word, ordered by date of first appearance. The y-axis is the sequence number assigned on the first use of an The black lines are fitted by three regression models, each over a range of days; the first and last lines have the form: $at\_num = a + b * (1 - e^{c*days})$, and the middle line has the form: $at\_num = a * days$; with a, b, and c constants fitted by the regression mod-

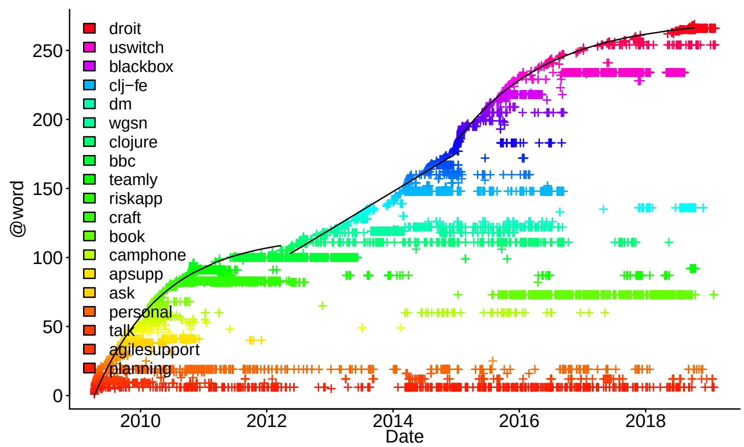

Figure 7: Time-line of first @word usage, with each @word assigned a numeric value, on first occurrence, appearing on y-axis; legend shows @words with more than 150 occurrences.

eling process.

Figure 7 shows several interesting patterns related to the use of "@word" tags. I decided to use tags to mark interesting aspects of tasks and tracking of work. Here are a few examples of classes of tags: Renzo

- Organisational tags: such as "@planning", "@meeting" or "@personal" denote activities related to the organisation of other tasks. They tend to appear across projects, and are distributed uniformly.

- Project tags: when a new project (or a consistent chunk of work related to a project) starts, it is assigned a "@word" descriptor. This helps me answer questions such as how long an entire project took start to finish. These tags are frequent when starting a new project and fade away within weeks or months. If many new word tags appear in the same period, it might signal an entire new job entirely, such as moving from one company to another.

Good productivity periods should show a pattern of cycling start/decrease/end of new project related tags. This signals focus periods followed by rest or refocus. Periods in which this pattern is not happening are likely to be less productive.

What is the rate at which new @words appear first in an @word list? The left plot in Figure 8 shows the number of new of new @words per 64 tasks (this choice gives the same number of points on the x-axis, as the left plot). Derek

An @word may be followed by a second @word that refines the task category specified by the first @word.

Figure 9 applies to the second @words, and was created in the same way as Figure 7...

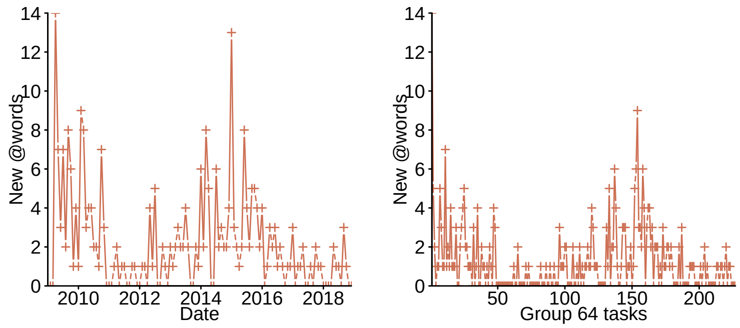

Figure 8: Number of first @words (in any task list) used for the first time in each month (left), and per sequence of 64 tasks (right).

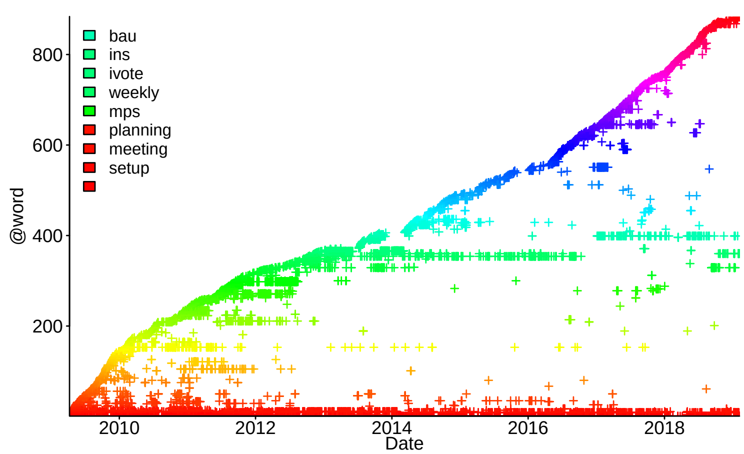

Figure 9: Time-line of second @word usage, with each @word assigned a numeric value, on first occurrence, appearing on y-axis; legend shows @words with more than 150 occurrences.

References

[1] F. Cirillo. *The Pomodoro Technique (The Pomodoro)*. Francesco Cirillo, 1.3 edition, June 2007.

[2] B. Flyvbjerg. How planners deal with uncomfortable knowledge: The dubious ethics of the American Planning Association. *Cities*, 32:157–163, June 2013.

[3] D. M. Jones. Code & data used in: Evidence-based software engineering: Based on the publicly available data. `http://www.github.com/Derek-Jones/ESEUR`, 2019.

[4] D. M. Jones and S. Cullum. A conversation around the analysis of the SiP effort estimation dataset. In *eprint arXiv:cs.SE/1901.01621*, Jan. 2019.