Draft: Uncertainties in Scientific Timelines

The closer you are to the finish line, the easier it is to plan a timeline. But a creative paper has a very long stretch where you are very far from the finish line, and it is very hard (probably impossible) to guess a timeline, because there are so many unknowns and unknown unknowns. The farther along you are, the more confidence you can have in the trajectory (like predicting a storm), but there are still many reasons that you might deviate from the original timeline.

Not all deviations or delays are created equal. Some are avoidable, some aren't; some unavoidable delays are still predictable, while some come out of nowhere. I've tried to catalog types of delays below, roughly ordering them from situations where you bear the least responsibility to situations that pose the most opportunity for personal growth. I then grouped them into two larger categories: Contextual Delays, which are not due to your actions, and Process-driven Delays, which reflect choices you've made about how to approach the work.

Contextual Delays

These are delays that are really outside your control. Now, an experienced scientist will anticipate that generic contextual delays will come up. They won't be able to predict the exact source of the delays, but they'll have a general sense that something will happen. So, a responsible timeline builds in some time for contextual delays. However, it's not realistic for students to have a good intuition for what is reasonable; because the data is stochastic, it takes experience to build intuition for it. Even a responsible timeline usually aims for an estimated mean time for "contextual delays", but half the time it might go over.

- Type 1: Nature Being Uncooperative

  Simulations aren't converging despite application of gold-standard methods; a new method fails in a corner case for reasons that are tricky to diagnose; the protein structure is collapsing; mechanisms are taking an unusually long time to reveal themselves.

- Type 2: External Dependencies

Someone or something else is late with their contribution: HPC is down; NAMD has a bug; collaborators aren't getting back to us; I get sick and cannot provide feedback or guidance. Note: does not include typical/reasonable queue or turnaround times or pre-announced downtime.

- Type 3: Low Availability

  You don't have as many hours as you reasonably expected to dedicate to the work, whether for unavoidable personal reasons like illness or due to unexpected higher-priority work tasks.

Duedates and timelines can be most concrete in cases where Type 1 & 2 delays are unlikely, like a course assignment or a presentation on established data. Any kind of R&D, on the other hand, is full of Type 1 & 2 delays. I am almost always willing to "slow down the clock" for necessary R&D. If I can't slow down the clock, I'll pitch in to help with R&D directly.

None of these delays are the student's fault, but just because something isn't our fault doesn't mean it doesn't cost us. It's not my fault if I get sick, but I still pay a monthly premium for health insurance (or in a different system, I would pay a higher tax rate). I'm also charged a copay if I see the doctor; the copay makes it less likely I will use the services unnecessarily. Even in most universal healthcare systems, I'd pay a small copay. Likewise, some PIs will cover the cost of contextual delays for their students, and others won't. Yours will. I provide very good R&D insurance, but I also sometimes charge a nominal copay. It reduces the temptation for frivolous R&D. That's economics, not judgment.

Process-Driven Delays

All scientists, but especially junior scientists, also fall behind due to process-driven delays. These stem from how the work is approached, through planning, perception, or internal response. These are more directly coachable and will improve as your professional instincts mature.

- Type 4: Inexperience or Miscommunication

  Honest mistakes due to inexperience; misjudging how long things take; misunderstanding directives.

- Type 5: Over-Refinement or Perfectionism beyond scope

  Rerunning simulations for marginal improvements; overdeveloping methods; rewriting already-clear prose.

- Type 6: Bad habits and Unforced Errors (We know better, but do it anyway)

  Basic technical slip-ups; failure to do expected quality control or sanity checking; failure to keep track of upcoming obligations that will affect your timeline; other violations of the commandments. Note: The same mistake might count as a Type 4 when you are junior (and don't know what you're doing) but Type 6 when you are senior.

- Type 7: Avoidance

  Procrastination; failure to initiate necessary tasks, communicate with colleagues, or engage with instructions.

Successful senior scientists have developed systems and practices to help minimize process-driven delays, but we all still suffer them. Still, the more control we can get over them, the easier our independent careers are. We face enough contextual delays; we don't have room for proliferation of process-driven delays.

As a PhD advisor, I'm obligated to help students learn how to minimize process-driven delays. If I just look the other way, I'm not doing my job. Now "Minimal" doesn't mean "none" — if "none" was the requirement, I couldn't be a scientist myself — but PhD students need to get them below threshold. To drive that change, I might try constraints, increased structure, or mild discipline, sometimes agonizing over the choice. I'll factor in the type of delay, the student's training level, and a lot of other circumstances.

Most Importantly

None of these categories reflect existential judgment. This is all just the Scientist's Serenity Strategy: accept the things we cannot change (contextual delays), and find the courage to change the things we can (process-driven delays). You'll hear from me about process-driven delays, but I think I would have to be really unreasonable to judge or reprimand graduate students for contextual delays. It would cause unnecessary pain for both of us, with no growth payoff.

Contextual-Delay Estimator (Grace's Intuition)

I'm going to try proposing an estimator that reflects my intuition. I don't make any claim that this reflects reality, but I think it does reflect my intuition about reality.

Consider a given task. If the best-case (lower bound) scenario is n weeks, this estimator provides an estimator for the upper bound.

$$\text{Lower bound:} \quad n$$
$$\text{Upper bound (95\% of the time):} \quad n + 2 \cdot \sigma_{\text{contextual}}$$

Where $\sigma_{\text{contextual}}$ is given by:

$$\sigma_{\text{contextual}} = \sqrt{\sigma_1^2 + \sigma_2^2 + \sigma_3^2} \tag{1}$$

Type 1 (Nature Being Uncooperative)

According to my intuition, the uncertainty due to Type 1 delays is

$$\sigma_1 = \frac{n}{4}\sqrt{(2nf_{\text{dev}})^2 + (1 - f_{\text{dev}})^2}$$

where f_{dev} is the fraction of the task that requires substantial development, and $1 - f_{\text{dev}}$ is the fraction that is straightforward. In the limit that $f_{\text{dev}} = 1$ (the entire task is R&D), then

$$\sigma_1 = \frac{n^2}{2},$$

while in the opposite limit of an entirely straightforward task, $f_{\text{dev}} = 0$ and

$$\sigma_1 = \frac{n}{4}.$$

Type 2 (External Dependencies)

I'll include the most common external dependencies we face, but if you have others, those would be added.

- HPC: $a \cdot n/2$, where the HPC factor a is a positive constant that increases as your cluster gets less reliable
  - $a = 1$ for a cluster of average reliability
  - $a > 1$ if it's prone to unexplained down-time or bad nodes
  - $a < 1$ if you rarely have problems
  - $a = 0$ if you're not using HPC at all
- Collaborators: $b \cdot n/4$, where b is the number of coauthors who need to have substantial input (other than the PI).

- PI Availability: $n/4$
- Software bugs: $n/4$ (I should probably give this a parameter based on the maturity of the software and the extensiveness of the testing process, but my intuition doesn't typically account for that. In practice, I haven't noticed that much variation.)

To get the total, again take the sum of squares:

$$\sigma_2 = \sqrt{\left(\frac{an}{2}\right)^2 + \left(\frac{bn}{4}\right)^2 + \left(\frac{n}{4}\right)^2 + \left(\frac{n}{4}\right)^2} = \frac{n}{4}\sqrt{4a^2 + b^2 + 2}$$

Type 3 (Low Availability)

My intuition predicts Type 3 delays almost entirely from "ambient stress", for two reasons: we are more likely to need personal maintenance when we are stressed out, and the same competing demands that will lead to delays also increase ambient stress. My sense is that this kind of uncertainty can be given by

$$\sigma_3 = \frac{c\,n}{8}$$

where c increases with "ambient stress", with $c = 1$ corresponding to a "typical" delay that adds 1 *unexpected* day per week to the upper bound. (If, under normal circumstances, you regularly lose more than 1 day a week to ambient stress, it should be built into your timeline: i.e. n should increase. Your "best-case" should reflect reality.)

If ambient stress is particularly low (say, you just got back from vacation) then set $c < 1$. If ambient stress is particularly high, set $c > 1$. How much greater? Estimate for yourself how many days a week (or weeks per month) you might unexpectedly lose to "ambient stress". That's your value of c.

Total Estimated Contextual Delays

Substituting all these terms into Eq 1:

$$\sigma_{\text{contextual}} = \frac{n}{4}\sqrt{\left(2nf_{\text{dev}}\right)^2 + (1 - f_{\text{dev}})^2 + (4a^2 + b^2 + 2) + c^2/2^2}$$

Examples

Scenario 1

Suppose the task is development-heavy ($f_{\text{dev}} = 1$) and there is no way it will take less than 2 weeks ($n = 2$). You're going to use a reasonably reliable cluster ($a = 1$), the only person you have to get it by is your PI ($b = 0$), and the ambient stress level is typical ($c = 1$):

$$\sigma_{\text{contextual}} = \frac{1}{2}\sqrt{16 + 0 + 4 + 0 + 2 + 1/4} = 2.4 \text{ weeks}$$

The 95% CI on the amount of time this task will take is

$$\text{Upper bound: } 2 + 2(2.4) = 6.8 \text{ weeks}$$

while the most likely time is

$$\text{Mode: } 2 + 2.4 = 4.4 \text{ weeks.}$$

That is, although it is possible you will complete this task in 2 weeks, it is most likely you will complete it somewhere between 4 and 5 weeks, and it would not be surprising if it took up to 7 weeks.

Scenario 2

Now, consider a task that is a very straightforward unbiased simulation,so $f_{\text{dev}} = 0$, and all the other parameters remain the same. Then

$$\sigma_{\text{contextual}} = \frac{1}{2}\sqrt{0 + 1 + 4 + 0 + 2 + 1/4} = 1.3 \text{ weeks}$$

The 95% CI on the amount of time this task will take is

$$\text{Upper bound: } 2 + 2(1.3) = 4.6 \text{ weeks}$$

while the most likely time is

$$\text{Mode: } 2 + 1.3 = 3.3 \text{ weeks.}$$

Scenario 3

Suppose you are writing up these results. However, a bunch of other tasks are piling up, so although it will take 3 full days to write it up, there's no chance it will be done before 5 days have elapsed ($n = 1$ week). This does not require development, so $f_{\text{dev}} = 0$, and it doesn't require HPC so $a = 0$. However, you do need to get it past 2 additional coauthors who are likely to have substantial comments ($b = 2$). Also everything is chaotic and you're a bit burnt out, so it wouldn't be surprising at all if you lost an additional 2 days ($c = 2$).

$$\sigma_{\text{contextual}} = \frac{1}{4}\sqrt{0 + 1 + 0 + 2^2 + 2 + 2^2/2^2} = 0.7 \text{ weeks}$$

The 95% CI on the amount of time this task will take is

$$\text{Upper bound: } 1 + 2(0.7) = 2.4 \text{ weeks}$$

while the most likely time is

$$\text{Mode: } 1 + 0.7 = 1.7 \text{ weeks}$$

[GB: I would not be surprised if there are some typos in the above. Let me know if you see some.]