

**Lisa  
Brown  
@Wertle**

## An Experiment...



Last year at Insomniac I led a small experimental project, wherein I had an insight about the relationship between accountability and efficiency. Since it was an experiment, we tried out some new processes – design logs, developer streams, and weekly builds.



Even though the goals of all these tools were different, they all had the same side effect: They consistently held my team accountable to different audiences of people, people who we very much cared about, and instead of the slog of adhering to an arbitrary deadline, the accountability made us more efficient

## Performance Pressure



Basically, if you can put yourself in a consistent position of having performance pressure to a trusted audience, that anxiety can be leveraged as a tool to shape your design process and make you more efficient. I'll talk about how we did it with the three tools I've mentioned.

# Tool #1: Design Logs



First, the design log was a living design doc idea lifted from an article by Spry Fox designer Dan Cook. It's a near-daily log of play notes, issues, and experiments that anyone can comment on, sort of a critical analysis of what is working in the prototype and why.



I decided to use the design log method for my solo prototyping phase.

Its original purpose was to communicate how the prototype was coming along to my leads and was supposed to be an afterthought to my work. But instead, the act of keeping the log started shaping my design process.

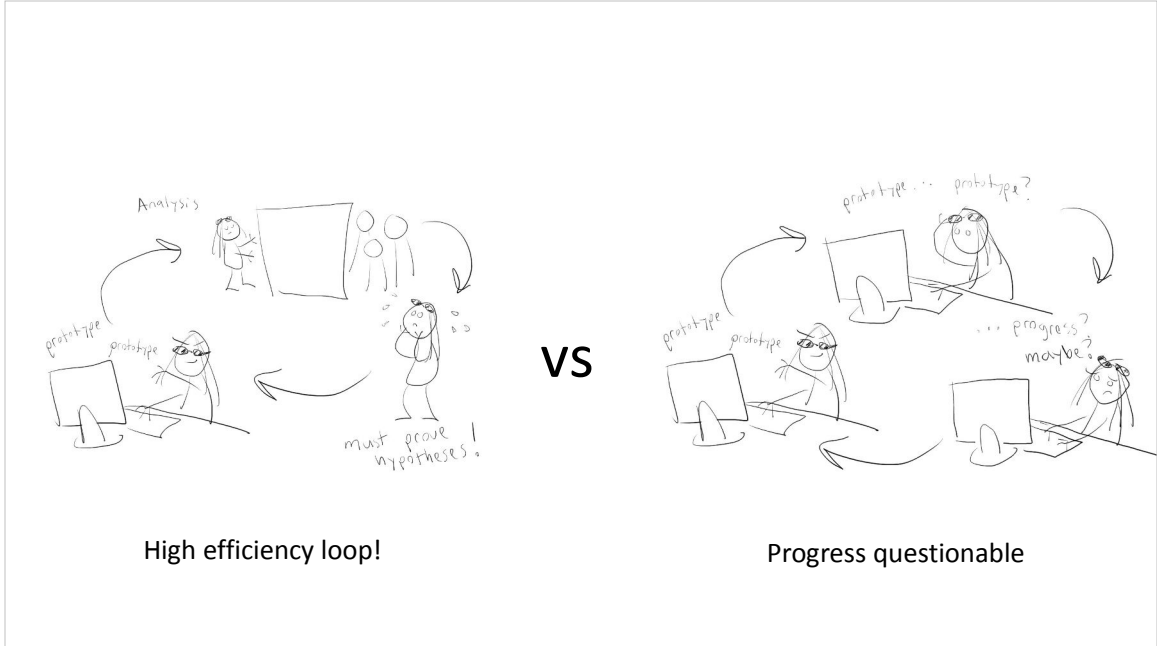


Knowing I would have to write one of these entries each day for the company leads started creating performance pressure! Pressure to address previous playnotes, test hypotheses written in my analysis, and to show that the rapid experimental nature of prototyping was still “progress.”



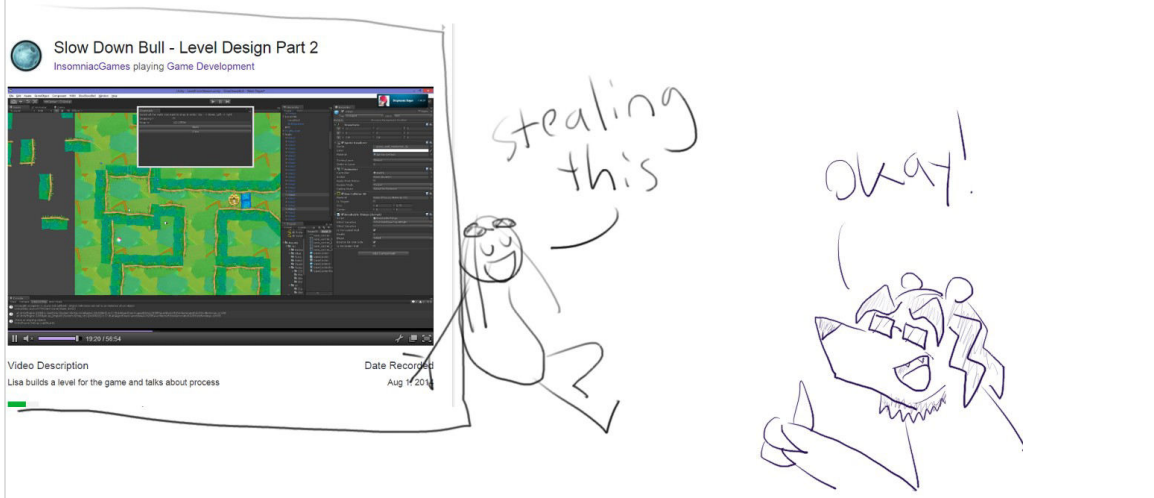
It made me design with the intent of justifying the trying of new things. Showing near-daily progress and having to critically articulate what was working and failing to people I cared about pushed me to be more focused on rapid, provable experimentation.



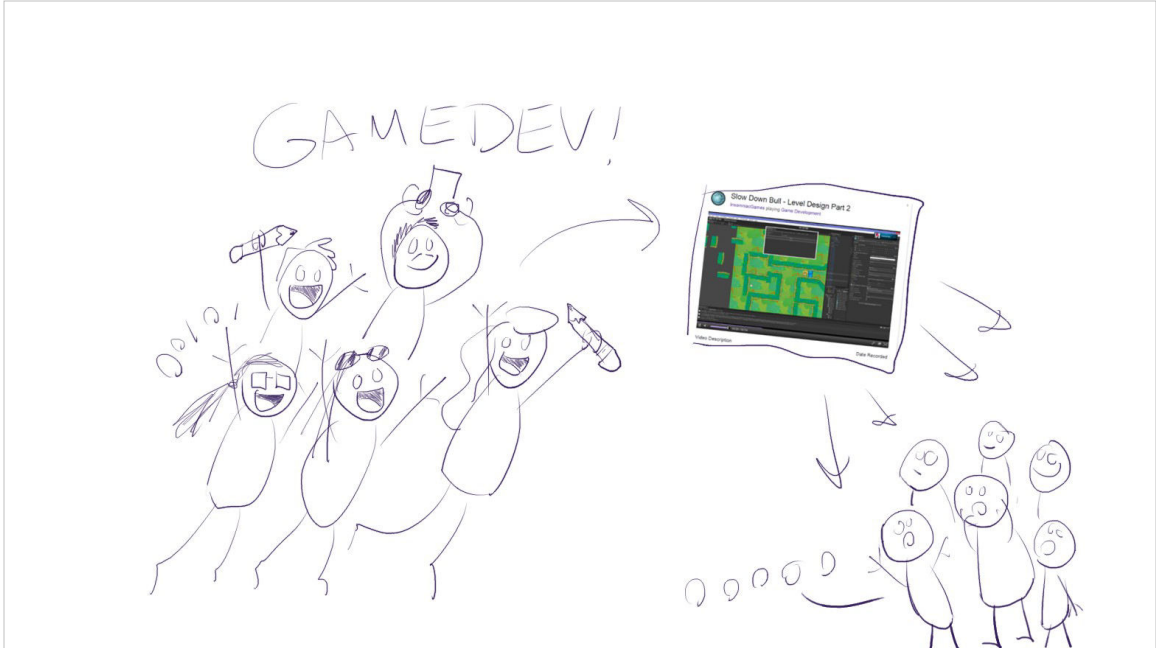


One might think “Lisa! These logs took a lot of time to write up! Would that time not have been better spent just prototyping more?” I don’t think so! I think the work spent writing the logs made a productive loop that yielded more progress over time, and kept me from wandering too much.

## Tool #2: Developer Streams



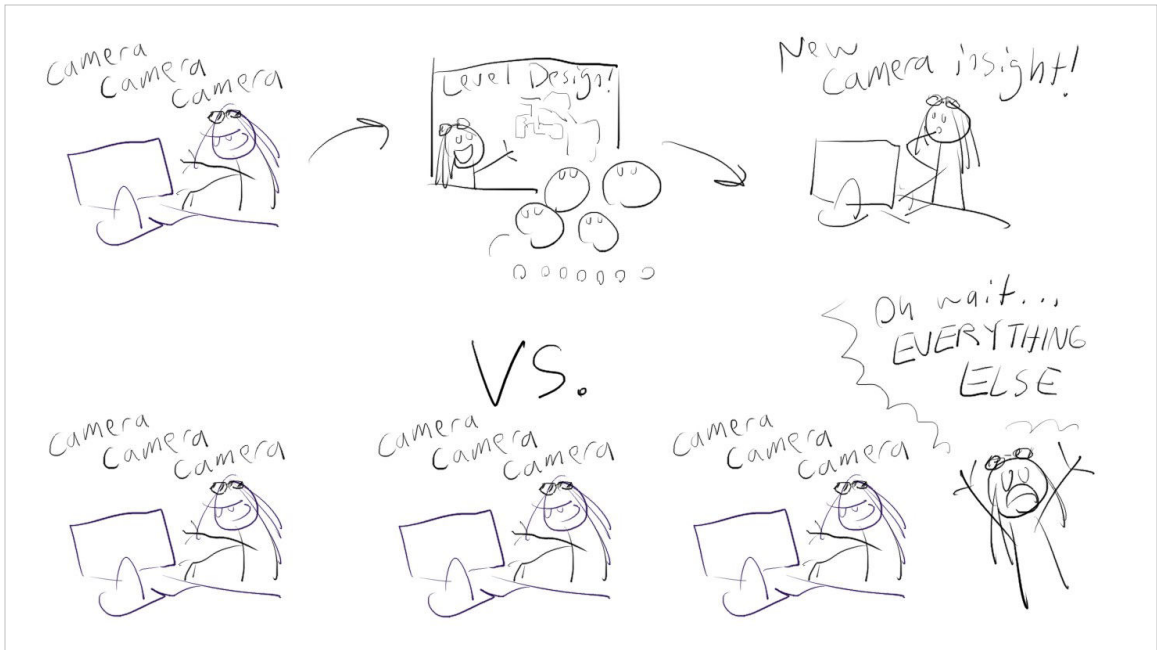
Next tool! Developer streams. By now we are in production and my team has decided to live stream ourselves working on the game once a week to an audience on twitch. The design logs were more academic in source but the inspiration to do dev streams came straight from the indies. (Thank you indies)



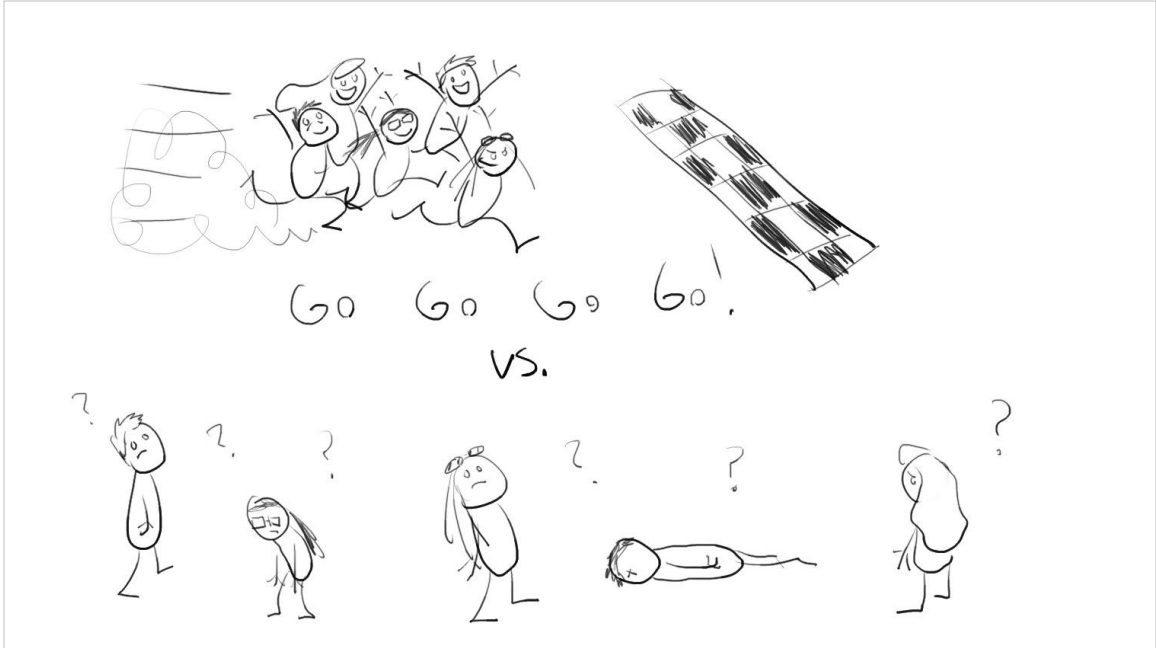
The original purpose behind streaming was outward facing, to experiment with format, foster a community around the game, and straight up educate our fans on the inner workings of game development. But after a few weeks of streaming I noticed something starting to happen with how my team was working...



The team had become acutely aware that we would be showing our unfinished, in-progress work to the public. Every week. All of a sudden, we were ON POINT for how efficiently we used our time. We scheduled in such a way that emphasized visible progress and held to it.



Design-wise, I became more efficient at tuning. Camera work, which I could have tuned for weeks straight, had to get to a certain threshold of “okay” before I temporarily shifted to something more watchable. This break cycle kept me from getting slogged down with tuning nitpicks. (and gave me more broad insights.)

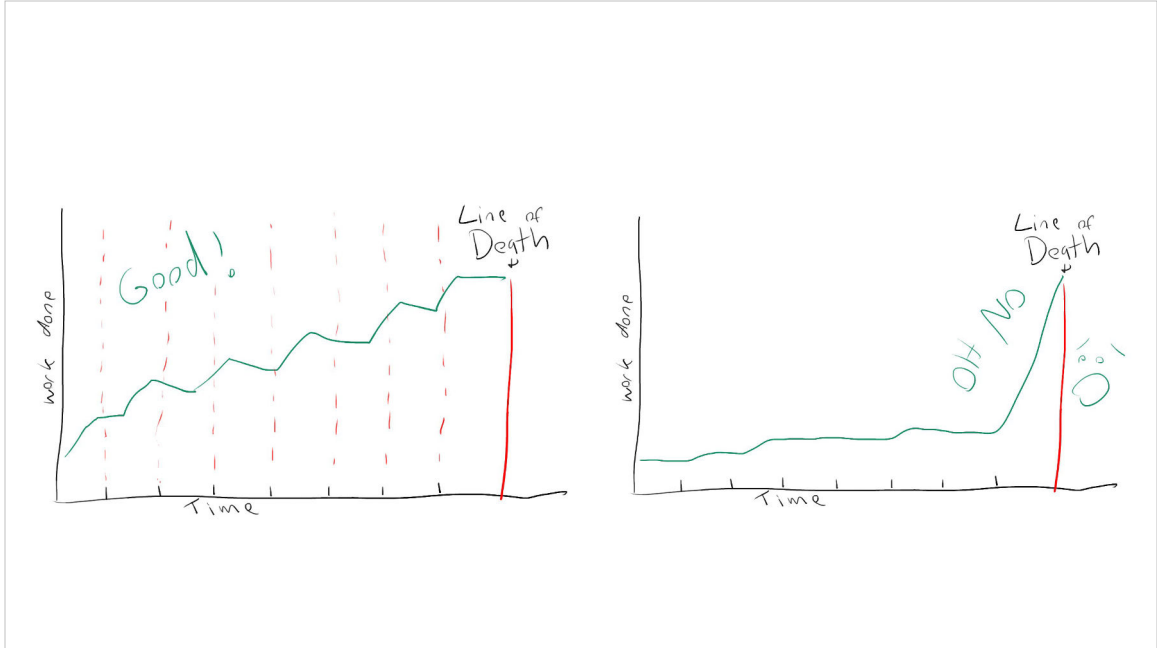


The streams were a fair amount of work to wrangle. But, like the design logs, I'm confident the extra work yielded more efficient results over time. We didn't have any of the demoralizing lulls that can happen in production and I'm certain its because of our accountability to show progress to our stream audience.

# Tool #3: Weekly Builds



End of project! Time for weekly builds. The logs came from the academics, the dev streaming from the indies, but the itemized weekly build checklist was straight up AAA, lifted from a process used on Sunset Overdrive that I wanted to experiment with on a small scale.

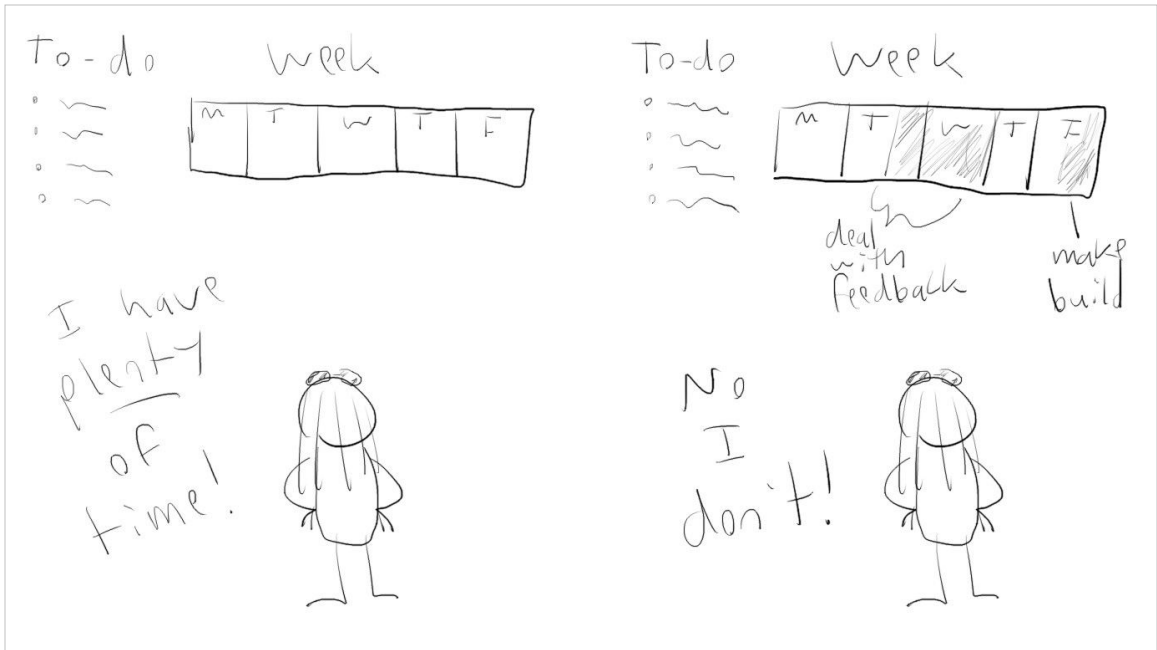


Scheduled with hyper-specific checklists for each build, the original purpose was to create intermittent milestones leading up to the end of the project while still making room to implement feedback, rather than a single looming deadline and the “oh shit” moment of being completely overwhelmed the week before it hits.





Knowing builds would be reviewed each week made me efficient at closing design loops. It's easy to think there's plenty of time to iterate on polish, but having itemized lists out to the final deadline adjusted with whatever didn't make it into THIS week's build makes you hit "this is good enough" REAL fast.



The reason accountability is important instead of just arbitrary weekly deadlines is that you're still forced to leave open time to assimilate feedback every week. This makes you commit to your final design decisions, so you won't waste time negotiating a suggestion when you know the answer is NO



The end-of-project push seems like the last place where you would want to spend a huge chunk of time responding to feedback, perhaps trying to figure out how to explain that your leads' suggestion is out of scope. But at this stage, losing that time makes you decisive and confident in your design calls.



So the biggest thing I want you to take away is that these tools only worked because they made us accountable to people that we cared about. We were not doing these consistent deliverables because we were forced to by some policy rule, but because we valued our audience's judgement and cared that they stayed updated.



It motivated us to show them progress, to NOT waste their time. There's a big difference between that and having to write an update every week because your boss told you to. Accountability alone won't make you efficient, but who you hold yourself accountable to and why you care about them.

**Lisa  
Brown  
@Wertle**