
Worker Intelligence

Alexandre Cela

About me

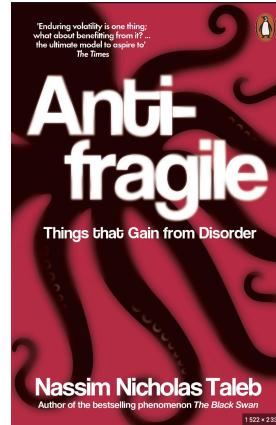
Big football fan, played in my Uni's Team



Big PSG fan, even though it's tough right now



*Decision making under uncertainty.
Favorite book, favorite author*



More generally, leveraging math, business acumen, instinct, knowledge, tech, AI to improve life's day to day, especially decision making

I love YouTube as I can learn anything there. Love the quite decentralised aspect of it as well

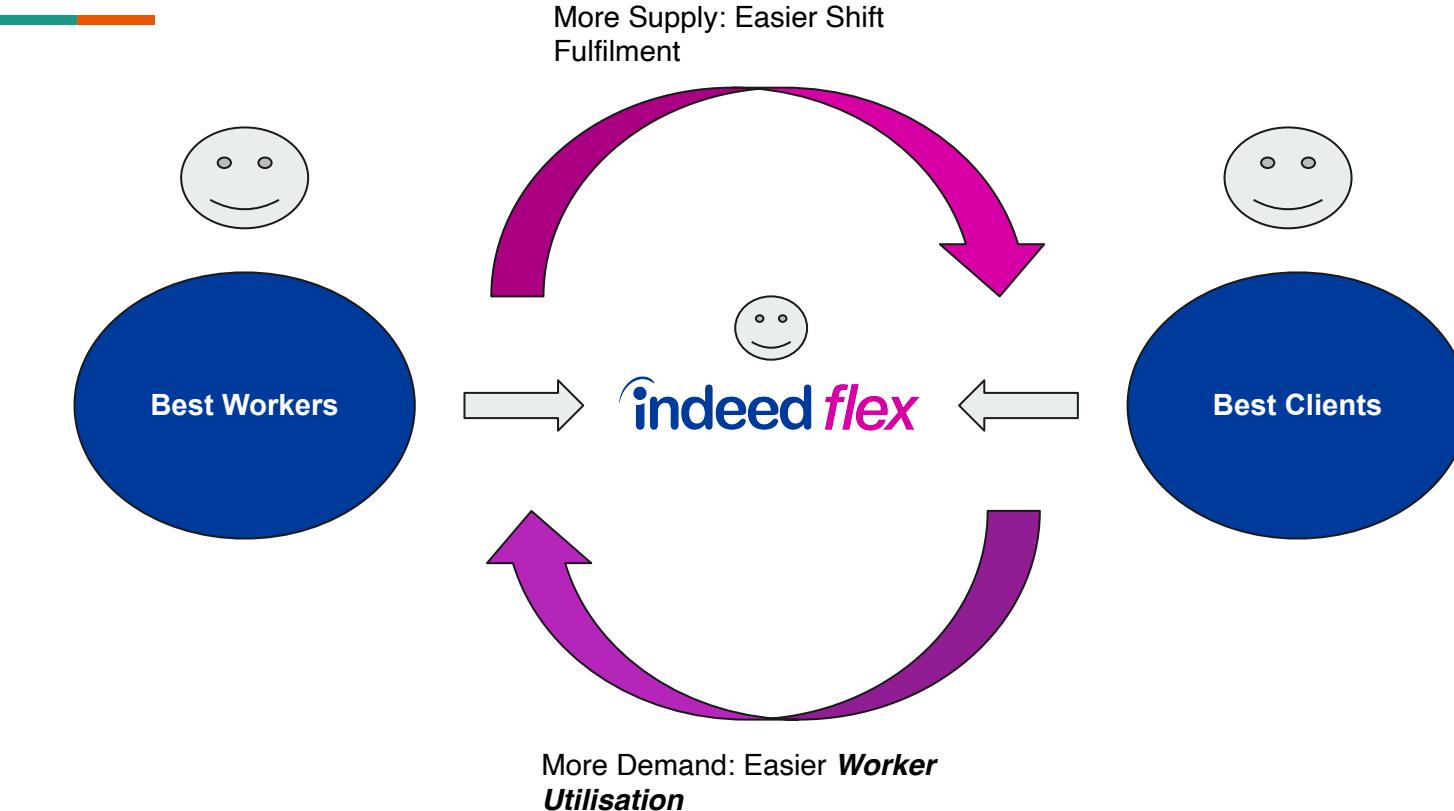


Project: Worker Intelligence



Data Platform leveraging technology, coding, business acumen, logic, to make day to day life easier at work, especially when it comes to gathering information, and making decisions

Context: IF Business model, and new focus on workers



— But who are our best workers? Do we know our workers?





Problem Statement

*And more generally... How to **Enable**, **Simple** and **Fast**, **Consistent** and **Reliable** key information consumption and research about our users?*



Enable

*As a Business Stakeholder, is there a **go to place** where I can quickly access **key user information**, such as who are our best users, how do they do over time. Who is trying to find work but is not. Who is not doing anything. Who is expected to find work, if so, how many? etc...*

*As an Analyst, is there a **go to place** where, for example, I can **quickly conduct research, hypothesis testing**, on which kind of job searching experience is increasing workers likelihood of finding work?*

Simplify, Fast

*Is there a **go to place** where I can **instantaneously** access **key user information**, in a **self serving way**, with just the use of a SELECT *?*

Let's avoid that

```
185 AS worker_id = d.worker_id AND shift_week = d.shift_week)
186
187     SELECT
188         *
189     FROM
190         activity_dsp,
191         rel_distance,
192         rel_activities,
193         shift_rel_ids,
194         sessions,
195         offer_rel_ids,
196         *
197     WHERE
198         activity.access >=
199             LTRIM(DATEADD(day, 5, d.worker_id))
200         AND activity.access <=
201             LTRIM(DATEADD(day, -5, d.worker_id))
202         AND shift_rel_ids >=
203             LTRIM(shift_rel_ids)
204         AND shift_rel_ids <=
205             RTRIM(shift_rel_ids)
206         AND rel_activities >=
207             LTRIM(rel_activities)
208         AND rel_activities <=
209             RTRIM(rel_activities)
210         AND rel_distance >=
211             LTRIM(rel_distance)
212         AND rel_distance <=
213             RTRIM(rel_distance)
214         AND sessions >=
215             LTRIM(sessions)
216         AND sessions <=
217             RTRIM(sessions)
218         AND offer_rel_ids >=
219             LTRIM(offer_rel_ids)
220         AND offer_rel_ids <=
221             RTRIM(offer_rel_ids)
222
223     GROUP BY
224         activity.access
225
226     ORDER BY
227         activity.access
228
229     SELECT
230         *
231     FROM
232         AS worker_id AS worker_id,
233         AS activity.access AS activity_access,
234         AS shift_rel_ids AS shift_rel_ids,
235         AS rel_activities AS rel_activities,
236         AS rel_distance AS rel_distance,
237         AS sessions AS sessions,
238         AS offer_rel_ids AS offer_rel_ids,
239         AS activity.access AS activity_intensity
240
241     WHERE
242         worker_id = d.worker_id
243         AND activity.access = d.shift_week
244         AND shift_rel_ids = d.shift_rel_ids
245         AND rel_activities = d.rel_activities
246         AND rel_distance = d.rel_distance
247         AND sessions = d.sessions
248         AND offer_rel_ids = d.offer_rel_ids
249         AND activity.access = d.activity_intensity
250
251     GROUP BY
252         worker_id,
253         activity.access,
254         shift_rel_ids,
255         rel_activities,
256         rel_distance,
257         sessions,
258         offer_rel_ids,
259         activity.access
260
261     ORDER BY
262         activity.access
263
264     SELECT
265         *
266     FROM
267         activity,
268         sessions,
269         shift_rel_ids,
270         rel_activities,
271         rel_distance,
272         offer_rel_ids,
273         worker_id,
274         worker,
275         worker_id AS worker_id,
276         activity.access AS activity_intensity,
277         shift_rel_ids AS shift_rel_ids,
278         rel_activities AS rel_activities,
279         rel_distance AS rel_distance,
280         sessions AS sessions,
281         offer_rel_ids AS offer_rel_ids,
282         worker_id AS worker_id,
283         worker AS worker,
284         worker_id AS worker_id,
285         worker AS worker,
286         activity.access AS activity_intensity
287
288     WHERE
289         activity.access = d.activity_intensity
290         AND shift_rel_ids = d.shift_rel_ids
291         AND rel_activities = d.rel_activities
292         AND rel_distance = d.rel_distance
293         AND sessions = d.sessions
294         AND offer_rel_ids = d.offer_rel_ids
295         AND worker_id = d.worker_id
296         AND worker_id = d.worker_id
297         AND worker_id = d.worker_id
298         AND worker_id = d.worker_id
299         AND worker_id = d.worker_id
300
301     GROUP BY
302         activity.access,
303         shift_rel_ids,
304         rel_activities,
305         rel_distance,
306         sessions,
307         offer_rel_ids,
308         worker_id,
309         worker,
310         activity.access
311
312     ORDER BY
313         activity.access
```

This incomplete, compared to the actual solution, query is not doing the job. It's a very long and complicated query, and the result takes too much time to show, or does not show

414 SELECT

{} {} LIMIT 1000 Save Execute

start_time	<input checked="" type="radio"/>	end_time	<input checked="" type="radio"/>	bad rating	<input checked="" type="radio"/>
2021-01-01	<input checked="" type="radio"/>	Today/Now	<input checked="" type="radio"/>	2	

Error running query: Query exceeded Redash query execution time limit.

Table SQL



Consistent

*How to make sure the information I am gathering is going to **be useful for quite some time**. How to be **robust against seasonality**, and to display information that will be **useful to as many stakeholder as possible, for as much time as possible**? Indeed, we would not want to need to change the query every week !*

We need to create a **objective framework, which breaks down the end to end experience of the user when interacting with our product**, in key different and successive steps, and compute metrics for each of these steps that give good information without providing too much noise

Having weekly data for every worker, **including when there is no data available for a given week for a given worker (a gap)**, would enable consistent information about churn or not, and be robust against seasonality



Reliable

How to make sure I can trust the data I see?

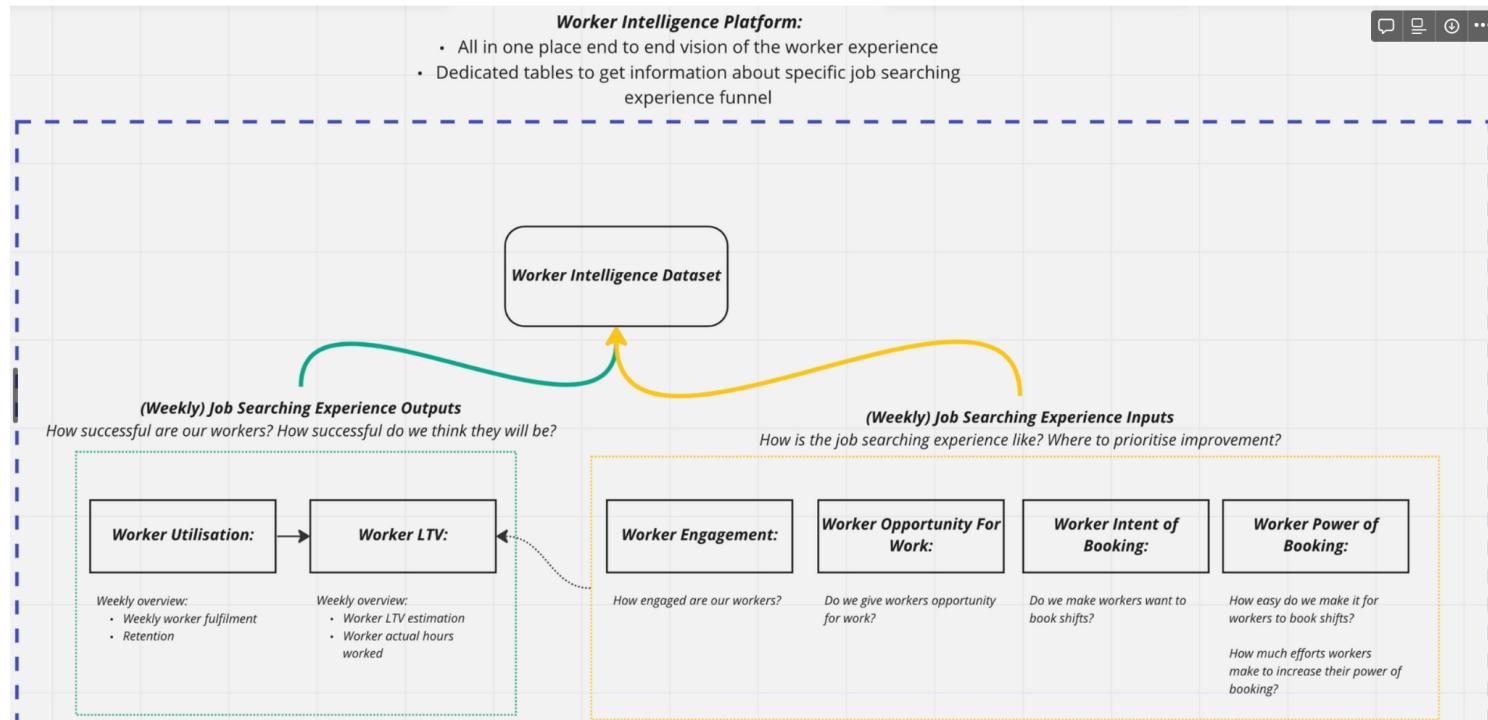
We would need to implement testing for every key metrics. This is not so easily done with only SQL

Solution: Worker Intelligence Platform, in DBT

- *An all in one* weekly dataset, combining 30+ raw tables, *enabling easy, fast, consistent and reliable key user information* :
 - Workers **output information**: How are our workers doing in terms of success?
 - Workers **input information**: What's going on with our workers, when they are looking for work? All above, for every (relevant) worker, and every (relevant) weeks. To provide a holistic view of the worker journey, and take seasonality into account.
- The platform is created in DBT through many SQL models, linked together to create the platform, which is then available for instantaneous querying in Snowflake. Thoroughly built with *Testing and Peer Review* (Git Hub), enabling *reliability*
- *Supercharging self serving research, EDA, hypothesis testing*

Solution: Worker Intelligence Platform Framework

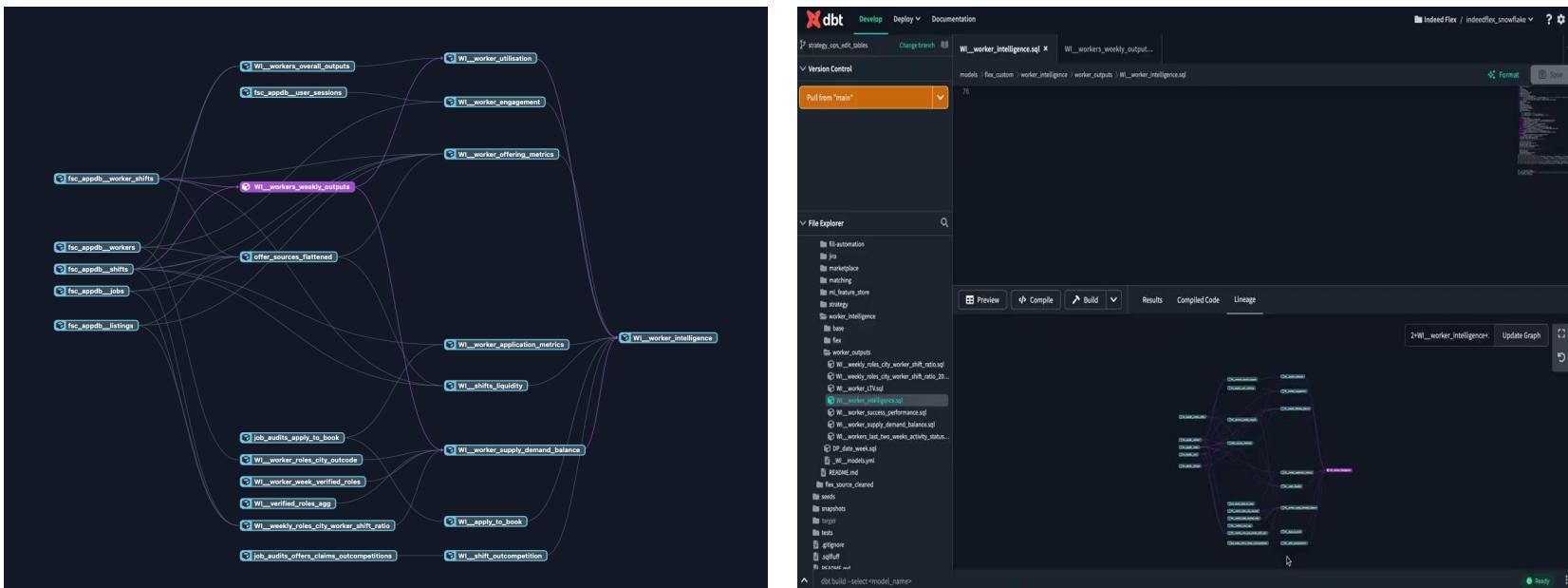
~~Consistency~~ thanks to the Framework. Precise but also general enough info. Framework can later be expanded



Solution: Platform enabled by DBT

Simple and fast, as well as reliable, information consumption, thanks to the (huge) work under the hood in DBT

Tree of the final dataset. Each of those bricks have their own lineage as well

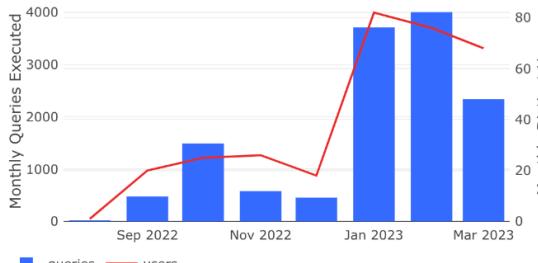


Results: Self served, fast, robust information consumption

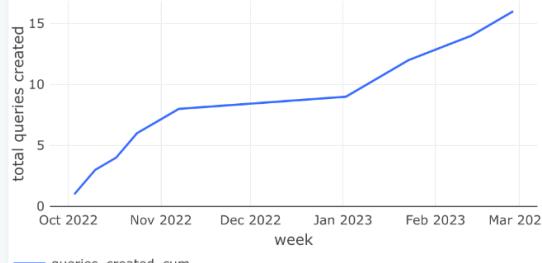
- Easy and fast **self served, robust** information access, for every stakeholder.
Even from business stakeholders. 5 sec max to wait
- Thanks to the framework, the metrics within the platform **cover most of the use cases / needs from PMs and business stakeholders**, which, even when they cannot self serve, **supercharges the dashboard creation time** allowing me to save more time for deeper analysis. Along with better tracking (e.g better WAU tracking)

The platform usage, by many stakeholders including SQL beginners, increases, and this with pending company wide communication

Worker Intelligence - Monthly Usage (exc. top user)



Worker Intelligence - Redash Queries Created (exc. top user)



Worker Intelligence - Who consumes the data? (last 30 days)

A table titled "Worker Intelligence - Who consumes the data? (last 30 days)". It lists names and the number of queries consumed. The table includes a navigation bar at the bottom.

name	queries
Adam Johnson	211
Alexandre Rodrigues	118
David Thompson	92
Annie Chen	92
Cerys Davies	87
Manu Fotedar	85

Results: Supercharging Data Analysis, Hypothesis Drawing, ML

- The data is almost a training data on its own. Combining success output metrics with input user experience metrics. In any case it makes easier to gather, over time, key features and use them for different purposes. The table is currently used for instance for a deployed LTV model
- Which I plan to trying to enrich, by adding more features from the user experience with our product. And it's super easy to analyse all features against success and draw hypothesis on which features can be useful for Data Science Use cases, or in any case to solve identified business problems
- No need to recompute again and again similar queries. And the query execution time is skyrocketed, reducing the feedback loop



Conclusion

- In short, it was very common that Senior Leadership people would insist on how we should shift more our business model towards our users and less the clients, but to me the actual resources needed for people to concretely be able to turn their attention more towards our users was not there
- It was almost impossible for business only people to access key and robust information about users, especially over time. It was possible but very time consuming, not scalable, for Analysts. Information was too scattered, some useful metrics had to be created, and making sure the data was okay added more time
- To this problem came my idea and the development of the Worker Intelligence platform, which, from DBT, outputs datasets in Snowflake which enables and simplify fast, consistent, and reliable key information about our workers. With just the use of a SELECT * or basic group by commands
- This platform is currently quite successful, especially without official communication. Business people are using it to self serve thanks to basic SQL, Analysts like me spend less time working on some dashboards for business people as the information is already there, which leaves more time to dive deeper and contribute to solving more intricate problems

Why I am proud about this project

- Identified a significant gap in the product, and took the initiative to fill it by ideating, and developing, on my own, this Data Platform. I had a bit of help from a Data Engineer in skilling up in DBT, and also designing the architecture and testing. I basically worked as both a PM and a developer on this fully fledged engineering product. ***I think having the ability to take a step back, identify key gaps and acting on it by owning projects and products should be relevant to the role here at Intuit***
- I created actual value for all the analytics community, by creating a go to place when it comes to consuming company wide user information. By shrinking massively the time needed to extract and analyse key information. PMs spend less time doing basic research, Analysts have more time to run more value added and deep analysis to solve business problems. ***For a company who aims at being even more data driven, I think being able to bring scale, and more ability to dive deep, are key***
- Even though the product has not yet been communicated company wide, there are many stakeholders using it already, including PMs with very basic SQL knowledge, which was one of the objective. ***The ability to create value and communicate this value efficiently, basically to be able to speak to business audience, is important***
- I learned many new things, including best practices when it comes to Data Engineering / Software Engineering principles. I learnt fast, while also maintaining my strong SQL coding capacities

Thanks