7 ways the Order to Cash process is crushing your Finance team, and what to do about it

Introduction

Order to Cash (OTC) is one of the key processes in every business, and drives a lot of finance's day-to-day.

But it's crushing Finance teams. Why?

1. Business models change faster and faster

One day, you discover discount codes are proliferating out of control. The next month, the assumption you made about dispute rates turns out to be way off, and you have to take a greater bad debt expense than planned, tanking the results. The month after that, you launched a new product with no visibility/segmentation to performance and with the launch engineering loaded a backfill that changed all your historical operational data. 😠

Today's digital businesses grow faster, and are more operationally flexible, than any that came before. They might start as a subscription business in one country, but within a few years, operate in dozens of countries with different payment infrastructures and currencies, and have additional marketplace or merchandise revenue streams.

A single ERP isn't set up to handle this much complexity and to change so fast. Like in many other business domains, technologists have unbundled once highly-rigid monolithic systems function by function into a vast array of software services. In fact, the ERP has been kicked out of the Order to Cash game completely. Rather than being the system of record for Order to Cash, it's just the system of final records for the general ledger.

2. Order to Cash got re-engineered, and Finance got left behind.

In order to fulfill the core function of offering goods and services and getting paid for them, engineering teams have rightly focused on stitching the Order to Cash process together with APIs and other integration methods.

But something absolutely essential **got left behind** in the process. In the urgency to adopt new payment processors, create new product functionality, introduce new business models, and more, the finance data produced in every step got completely disconnected, just like the operations data. And while engineering rightly prioritized connecting the operations data together to run the business, Finance was largely left behind to figure out how to connect the dots. And that has made for an unholy mess.

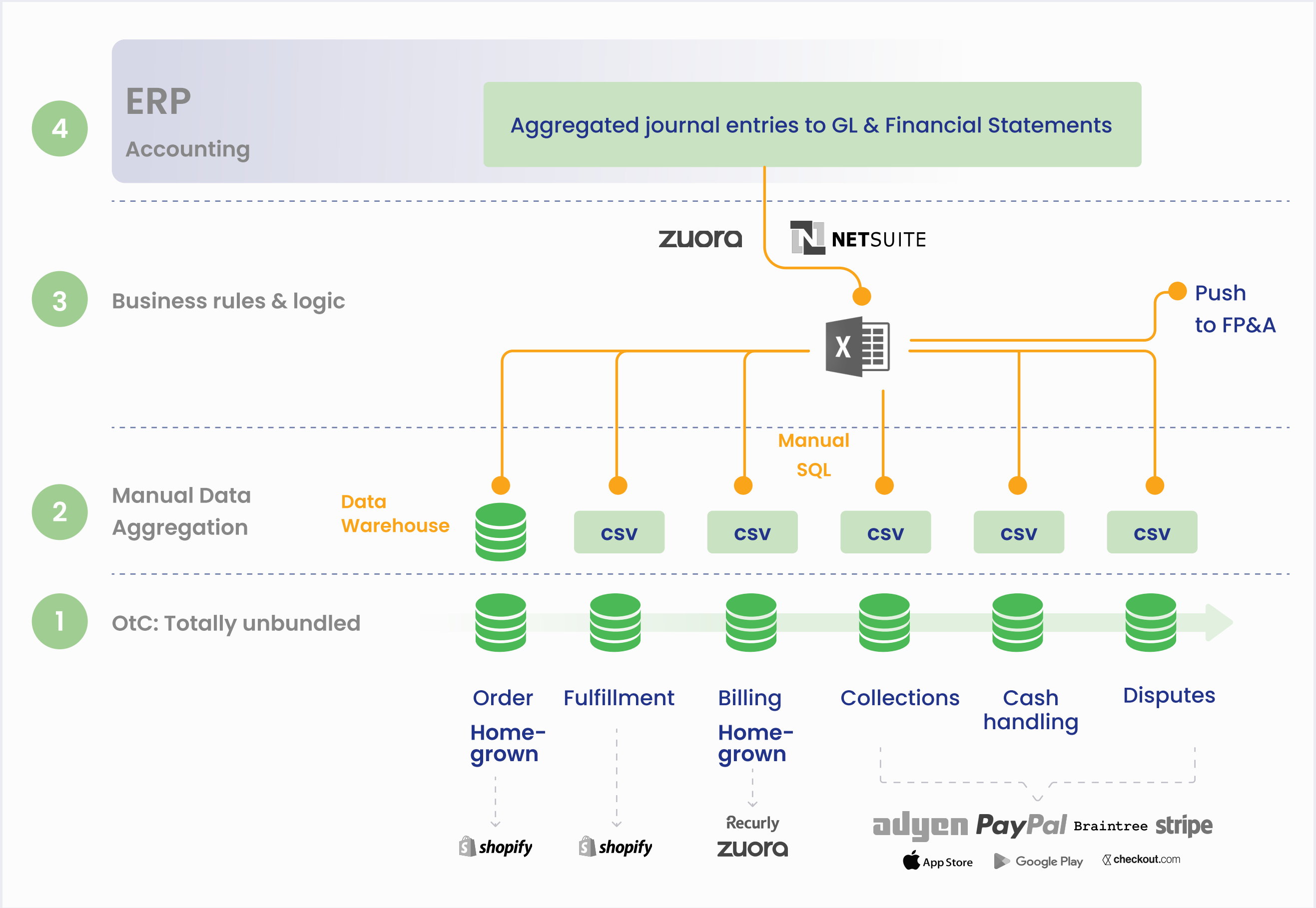
Imagine a basketball game where every player keeps their own score, rebound count, and shot clock, and, at the end of the game, the scorekeeper collects slips of paper with illegible handwriting from every player with their self-reported stats and tries to figure out the final score. That's the challenge that finance teams face figuring this all out.

Now imagine a tournament where, as soon as one game is finished, the next one starts. But there's still only one scorekeeper. So during each game, they're frantically trying to finish figuring out the score for the last game and can't pay attention to the current game right in front of them. And the players from the last game are getting antsy because they want to know the result and which team to prepare to play next based on the bracket.

That's one stressed out scorekeeper!

The Finance team, which desperately wants to be engaged in the game right in front of them — to coach, to referee, and even to play — is so consumed trying to reconcile the stats for the last game, they never have a chance.

Here's what Finance's reality looks like now. It's not the tidy all-in-one Order to Cash process they keep trying to build:



1 OtC: Totally unbundled.

2 Data lineage broken with manual data aggregation.

3 Brittle business rules & logic applied manually in Excel.

4 Only aggregate data pushed to GL; ERP scope only GL

FINANCE IS STUCK:

- Lack detailed visibility
- Slow
- Brittle risk for material errors & audit failures

> Each operational system Product and Engineering teams use to support high-volume transaction processing — *from taking an order, accepting a payment, fulfilling and shipping, handling disputes, and so on* — only tells a part of the transaction story.

3. Reconciliation has become an unholy mess

In the past, reconciliation was pretty easy — it mostly happened automatically in the ERP. The whole OTC process was tied to one data model, tracked in a single ERP system. Running the actual processing of orders, fulfillment, payments, collections, and cash was the game, and one a month, Finance powered the scoreboard.

Today, reconciliation is so much harder.

Instead of one data model residing in one system, the complex web of homegrown or standalone billing and order management systems, numerous payment processors, and app stores has exponentially increased the number of data sources that Finance teams have to reconcile.

APIs are everywhere, emitting data from a slew of disparate sources and structures in the tech stack:

- Enterprise resource planning (ERP)
- Order management systems (OMS)
- Customer relationship management (CRM)
- Payment providers (PSPs)
- Accounts payable system (APS)
- Business intelligence (BI)
- Enterprise data warehouses (EDW)
- HR systems (HRIS)

On top of that, there are all the other data sources, like your bank and any internally developed software that emits financially relevant data, which is basically all of them.

4. Reporting and audits have gotten insanely hard, too

To generate financial reports, Finance teams need to extract the relevant data from these systems, often downloading huge CSV data files one by one, tie and reconcile it all together, analyze it, and then use it to build reports. The business logic they're using to generate journal entries isn't in a standardized, controlled system; it's in innumerable spreadsheets, one typo away from a disaster.

With every additional transaction data nuance and every new data source, accountants risk making more manual errors, struggle even more to trace line items back from financial reports to source data, and undertake greater contortions to reconcile accounts and close the books. As they get pulled deeper into spreadsheet acrobatics, financial analysts get farther away from advising and driving business.

These challenges slow business decisions and month-end close, reduce confidence in reporting and projections, and create unholy levels of stress when auditors ask to see how the reporting relates to the source data from the operational systems. In turn, Finance is left with no time to devote to higher value work to root out waste, optimize accounting policies, or strategize with the business leaders about deploying financial resources to boost the bottom line.

Put yourself in the shoes of the Controller in the midst of an audit, and they get asked:

- Why did last month's numbers change each time you pulled the data?
- Why does your GL back up from the prior month not match the source system?

- How did you generate this journal entry from fulfillment data, shipping date, and dispute information from three separate data sources?
- These are terrifying questions.

5. It’s harder to serve the business

Perhaps worst of all, all the manual work — to reconcile, to close, to audit — means you aren’t maintaining the data at a transaction level and lose fidelity and detail with every step of the process. Detail that — if you had it at your fingertips — would be enormously valuable to optimize the business. For example:

- Finding customer segments that have higher return or bad debt and avoiding them.
- Choosing the right shipping partner and cost estimates for every shipment to stay ahead of dynamic market conditions during the all-important holiday season in ecommerce to prevent margin erosion.
- Tweaking product offerings or pricing to encourage customers to choose more profitable products.
- Pulling forward revenue by recognizing line item revenue recognition in lieu of proxy fulfillment dates or the entire order being fulfilled.

It’s impossible to do these types of analyses accurately without a complete view of every single transaction. And in today’s world of unbundled Order to Cash and manual finance acrobatics, that view is just plain missing.

No wonder you feel frustrated. You're stuck working harder and harder each month while delivering less and less value.

6. Engineering a solution to the problem is possible…but hard

Let's imagine, for example, that you need to understand net profit per order with orders in your ERP, payments in your PSP and selling expenses in your HRIS. To engineer that into your data pipeline you need to write code to make a call to your ERP API that extracts data with a REST connector/authentication/validations API to get all the data in a JSON payload and load it into a central data store.

But you also need to tie variable commission expenses to your orders to get to net profit, and that data lives in your HRIS/Compensation platform, not your ERP.
Unfortunately, that API requires us to send SOAP requests and now you're engineering a completely different authorization process which you then have to serialize from XML into JSON. Lastly, your PSP has a native integration to your ERP but you spend more time tracking down and re-running failed transactions, entering in missing records manually and correcting transactions into their proper period. Instead, you go ahead and pull a raw data set directly from the PSP into a CSV and upload to your data warehouse as a one time backfill.

That's just a rudimentary example using only data formats from common APIs and without touching on data structures and transformations to get you to the end state of the actual report or semantics layer where analysts can actually find value in the data! Performing gymnastics within these complex layers of your data stack all start to look like alphabet soup and takes you further away from getting the data you need quickly to guide critical business decisions.

7. Not only is the data complex — you're also drowning in it

Excel works well enough when you're trying to process small amounts of data. But soon enough you find your Excel models crashing, taking 20 minutes to open, or split into 37 files.

For digital businesses with high transaction volumes, Excel is no longer viable and programmatic data processing methods also get stretched beyond their limits causing data processing lags or failures.

For small datasets, it's not uncommon to build pipelines using in-memory processes (like building Pandas dataframes in Python and loading them into a data warehouse like Snowflake).

But as your datasets grow, you will hit a point where your infrastructure can't handle the load (either by taking hours to run DAG pipelines or running out of memory entirely) and you need to move to a distributed process. And, forget about capturing changed data on massive data sets.

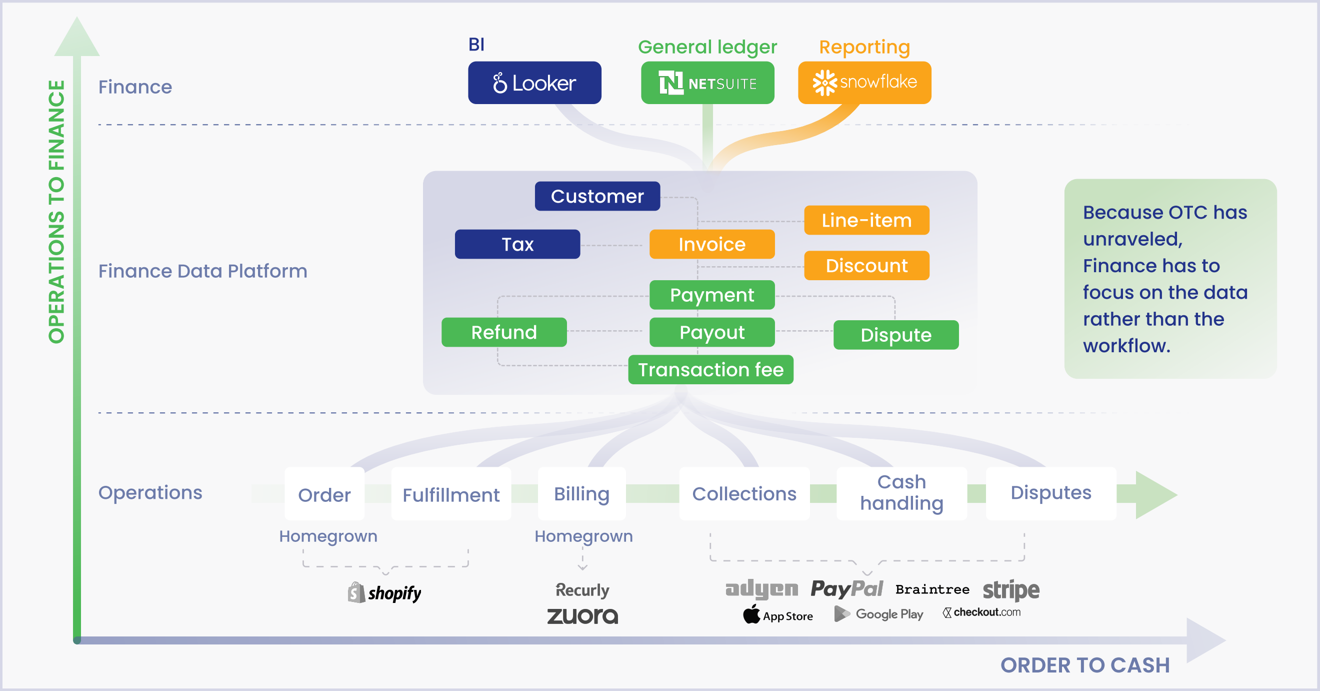
That's a whole other headache! Now if you're processing data in your ERP and Excel, data is growing non-linearly so hitting limits — on the API, concurrency, or on your contractual rights, or exceeding the Excel limits for a monthly close, isn't at all uncommon. With historical operational data constantly shifting, you have no choice but to re-run and do a variance analysis across all historical accounting periods to know you've captured all data changes. That's just one example of what drives this non-linear growth.

Exponential growth will hit you in the face much faster than you might expect. Can your ERP handle it? Sure, they'll take your hundreds of thousands of dollars. Do you have that to spare?

Seven challenges, One solution: stop optimizing processes, start optimizing data

It's time to recognize the incredible disconnect between the beautiful vision of Order to Cash you've been working toward for thirty years, realize that it's no longer viable, and shift your mindset.

It's time to pivot your thinking from a horizontal view of Order to Cash – a view focused on process — to a vertical view, oriented around the data flows that are causing you so much heartache. In the vertical view, the focus is on taking operational data and translating and transforming it into finance data.



Because OTC has unraveled, Finance has to focus on the data rather than the workflow.

From OTC to OTF

If you shift your approach from Order to Cash and pulling financial data at the end of a period to a new strategy of real-time data, you unlock a whole new world of possibilities..

It's how you can free yourself from the incredible weight of the scorekeeping process you currently manage, and unleash your analytical and strategic finance firepower to lead strategic business change. It's how you go from Finance being an afterthought and consequence of the operational activity in the business, to being a driver of the choices that lead to operational success. It's how you go from being reactive to leading the business.

In today's market, Finance can't afford to be on the sidelines anymore. High growth digital companies today no longer have the luxury of seemingly unlimited and free capital. Finance leaders have to make sure every dollar is spent judiciously. You have to find pockets of profit degradation from operational decisions made in a vacuum,
and you have to do it immediately.

Join us on the OTC to OTF journey, read more about how to adopt an OTF process **here**.