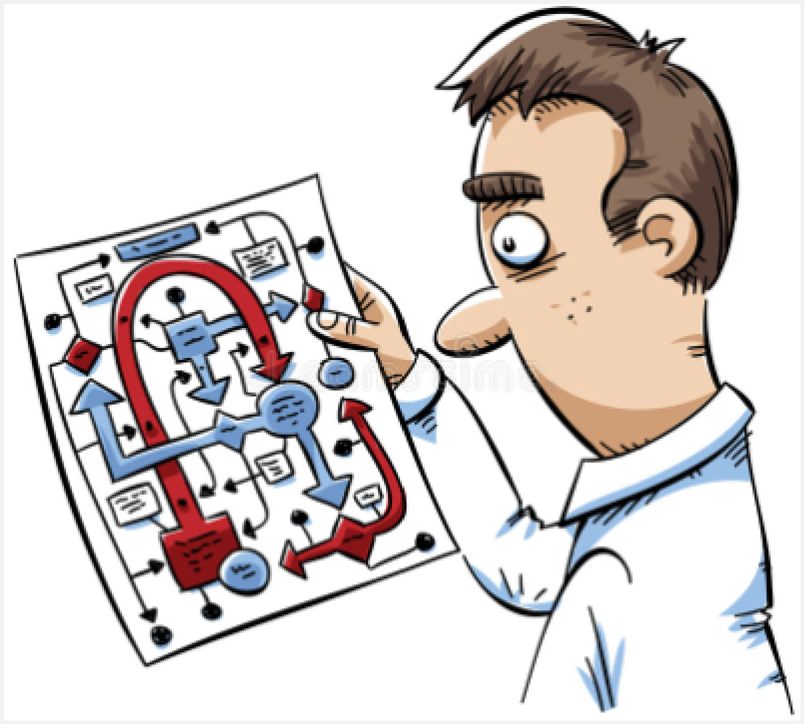
BUZZ CHRONICLES > ALL
Saved by @sachm0
See On Twitter

Twitter Thread by Frank Rotman

Frank Rotman
@fintechjunkie

1/16: One business model I talk about frequently with Founders is underbuilding their software as a strategy.

It’s a really powerful concept that can help a product stand out in a crowded market and turbo-charge growth. Unpacked ■:



2/16: It’s difficult to deny that just about everyone is a user of software on a daily basis. Phones and Computers are just Operating Systems + Pre-Loaded Apps + Downloaded Apps. Apps include streaming services, browsers, spreadsheets and POS software. The list goes on and on.

3/16: But ask yourself this: How many functions do you typically use in the software/apps that you interact with most? Streaming services: Search, create playlists, play. POS software: Add item descriptions and pricing, ring up orders, view reports.

4/16: The concept of purposefully underbuilding software hinges on the fact that there are situations where doing a few things extraordinarily well can be more valuable to an end-user than offering a complete suite of products/functions that are all “pretty good”.

5/16: Truism #1: Solving for all common use cases is more difficult than solving for a specific use case. Solving for edge cases is even more difficult. Going deep with a few pieces of specific functionality is a way of underbuilding to create advantage.

6/16: Example: Building a CRM for a VC firm is different than building a CRM for a Real Estate Agent is different than building a CRM for a B2B sales organization. Specialists are emerging in various verticals designed to deliver limited but industry specific functionality.

7/16: A major advantage is that “built for X” is a marketing message that has tangible value. There are channel strategies, content strategies and sales strategies that can be anchored around this message and the response rate and funnel advantages can be significant.

8/16: So while the 800 pound gorilla in the CRM space is @Salesforce, it’s a fully functional solution that needs to be configured for each use case. This can be intimidating, expensive and requires lots of input/expertise from the buyer who doesn’t always know what they need.

9/16: Truism #2: Limited functionality makes everything easier for the end user. They can research and compare options easier. They can visualize using the product easier. They can be onboarded and get started easier.

10/16: This reduction in friction has tangible value. Site visits will convert at a higher rate. Time with expensive sales agents can be reduced/replaced with collateral like downloadable PDFs and 90 second explainer videos. The “sign-up but never use” segment can be eliminated.

11/16: Another massive benefit is that limited functionality focuses a customer on using the product the intended way which sets them up for success. More customer success = Higher NPS = Higher retention + More referral growth + More long-term pricing power.

12/16: Truism #3: Designing a world class UX/UI for a product with limited functionality is much easier than it is for a product with dozens of features. This has always been true but even more so now that tablets and cell phones have become dominant input surfaces.

13/16: Building the mobile experience first is a smart strategy that will challenge a product team to simplify scope due to limited surface area and clunky input devices (like fingers!). Doing this for a product with feature overload is incongruent with mobile constraints.

14/16: But a major advantage of having a fantastically designed mobile experience is that engagement with the product is almost always greater than on a laptop/desktop. Mobile devices are untethered and never out of sight. More engagement = More perceived value.

15/16: The TL;DR is that very few users of products/software value more than a small handful of features and therefore underbuilding the product can have tangible value. Adding features can expand the market of buyers but will also make everything else more difficult.

16/16: And while there are many situations where a product needs full functionality (including edge cases) to be buyable by users, this isn’t universally true. And where it isn’t true, underbuilding a product can be a smart strategy.