MVP Features: What Went Into 43 First Versions We Built

Alpit Patel, Founder and CEO, aibrevo (San Francisco) · Published November 14, 2026

ABSTRACT

We studied the first launched versions of 43 software products built by our in-house team and coded the 247 headline features in their published case studies into 19 building blocks. A typical first version had 5 blocks (median; range 1 to 10). The most common were an operator dashboard (23 of 43), payments (21), notifications (18), booking or scheduling (17) and search (16). Marketplaces (16 builds) differed sharply from other products: 75% took payments, 50% verified their providers and 31% held money in escrow, against 33%, 7% and 0% elsewhere. Among the 41 apps, connecting to an outside system was the block most consistently linked to longer builds, adding 0.9 weeks (permutation test, p = 0.008). Taking payments added no time (difference -0.1 weeks, p = 0.77). The number of blocks did not predict build time (r = -0.02). We conclude that MVP timelines depend on which blocks are chosen, especially dependencies on other people's systems, more than on how many features are listed.

Keywords: minimum viable product, MVP features, software scoping, marketplace, build time

1. INTRODUCTION

The question we hear most from founders is what has to be in version one. Most published answers are opinion. We had first-party records instead: 44 products we built, each with its feature list, build time and results written up as a case study. This paper asks three questions. What do first versions contain? Do different product types need different blocks? Which blocks make a build take longer?

2. METHOD

2.1 Data

The source is our 44 published case studies. One (DAIN) has no feature list and was excluded, leaving 43 builds and 247 headline features (5 or 6 per build; the cap is our writing format). Build time is the reported time to the first launched version, in whole weeks; most case studies count from kickoff, a few from the first call or after a separate scoping week. 16 builds are marketplaces; 2 are websites rather than apps.

2.2 Coding

Each feature was coded into one or more of 19 building blocks, using only the full text of the feature list. The blocks are: operator dashboard; payments; held payment (escrow); verification of users or providers; search and filters; matching or automatic routing; booking, scheduling or calendar; reviews;

notifications; direct messaging between users (comments and group threads count as social); connection to an outside system; documents, contracts or e-signature; map or location; AI or machine-learning feature; social layer; approvals or roles; export; offline operation; self-managed content. Coding was done by one person.

2.3 Analysis

We counted blocks per build and across builds, compared marketplaces with other products, and compared mean build time with and without each of the 8 most common blocks. Significance was tested with a two-sided permutation test (20,000 shuffles). Time analyses use the 41 apps only, since both websites are short builds.

3. RESULTS

3.1 Frequency of building blocks



Figure 1. Number of the 43 builds whose published feature list includes each block.

The median build had 5 blocks (mean 4.8, range 1 to 10). More than half of all first versions named an operator dashboard as a headline feature. Direct messaging was named by 3 builds, offline operation by 2 and AI by 4, in 3 of which AI was the product itself.

3.2 Marketplaces compared with other products

Building block	Marketplaces (16)	Other products (27)
Payments	12 (75%)	9 (33%)
Search and filters	10 (63%)	6 (22%)
Booking or calendar	9 (56%)	8 (30%)
Verification	8 (50%)	2 (7%)
Held payment	5 (31%)	0
Reviews	5 (31%)	1 (4%)
Approvals or roles	1 (6%)	10 (37%)
Outside connection	4 (25%)	11 (41%)

Table 1. Share of builds naming each block, by product type.

3.3 Building blocks and build time

Building block	Apps with it	Weeks with	Weeks without	Difference	p
Outside connection	15	7.7	6.8	+0.9	0.008
Documents	12	7.7	6.9	+0.7	0.05
Matching	10	7.5	7.0	+0.5	0.24
Operator dashboard	23	7.3	7.0	+0.3	0.47
Map or location	12	7.3	7.1	+0.3	0.52
Payments	21	7.1	7.2	-0.1	0.77
Search and filters	16	6.6	7.5	-1.0	0.005
Self-managed content	8	6.4	7.3	-1.0	0.02

Table 2. Mean weeks to first version for the 41 apps, with and without each of the 8 most common blocks. p from a two-sided permutation test, 20,000 shuffles.

The longest build (Spirit Airlines group booking, 10 weeks) connected to the airline's fare system. The 7 apps with both payments and an outside connection averaged 8.1 weeks. The 17 apps with neither an outside connection nor documents had a median of 6 weeks. Shorter times for search reflect the type of product (discovery apps over the product's own data) rather than an effect of search itself.

3.4 Number of blocks and build time

Across the 41 apps, the correlation between the number of blocks and weeks to launch was -0.02 (0.17 with the two websites included). One build mapped to a single standard block and still took 8 weeks; its effort sat in product-specific logic (adaptive learning paths, branching mock interviews, an in-browser modelling test) that no category list captures.

4. DISCUSSION

Our reading is that a first version should be planned around dependencies, not feature counts. Payments, a standard block, cost nothing extra in time; a connection to a system someone else controls cost about a week, because its documentation, test access and response times sit outside the team.

Product type sets the shape of the list: marketplaces need trust between strangers on day one (payments, supply-side checks, held money), while internal and B2B tools need approvals, roles and links to existing systems. Several builds also show that scope is a set of bets rather than a frozen list: one marketplace left out fee line items by design, one tracking app reduced daily logging to a single screen, and features added mid-build (offline cash logging, audio recitation) became important to users.

These results fit wider evidence for keeping version one small. Pendo found that 80% of features in the average software product are rarely or never used [2], and CB Insights, studying 431 VC-backed shutdowns since 2023, attributes 43% of the 385 with an identified cause to poor product-market fit [3]. An MVP in Ries's sense, the version that yields the most validated learning for the least effort [1], is the practical response to both.

5. LIMITATIONS

- First-party data from one team; all builds launched, so the set is tidier than a full project history.
- Headline features only: a block missing from a feature list may still exist in the product.
- A single coder; the full coding is published so it can be checked or redone.
- Small samples and 8 time comparisons; results near p = 0.05 (documents) should be read as indicative.
- Build times are whole weeks, and the starting point varies slightly between case studies.

6. CONCLUSION

A typical MVP in our data had 5 building blocks. Include an operator view, take payments from day one if the product will charge, budget about an extra week when the build depends on an outside system, and leave messaging, AI extras and export for later unless the product is about them.

DATA AVAILABILITY

The coded dataset (one row per build, one column per block) is available at aibrevo.com/research/mvp-features-43-builds.csv. Each build's case study is at aibrevo.com/work/.

REFERENCES

1. Ries, E. (2009). Minimum Viable Product: a guide. Startup Lessons Learned. https://www.startuplessonslearned.com/2009/08/minimum-viable-product-guide.html
2. Pendo (2019). The 2019 Feature Adoption Report. https://www.pendo.io/resources/the-2019-feature-adoption-report/
3. CB Insights (2026, March 5). The top 9 reasons startups fail. https://www.cbinsights.com/research/startup-failure-reasons-top/

Cite as: Patel, A. (2026). MVP Features: What Went Into 43 First Versions We Built. aibrevo. https://aibrevo.com/research/mvp-features-43-builds/