Building Developer-Driven Platforms

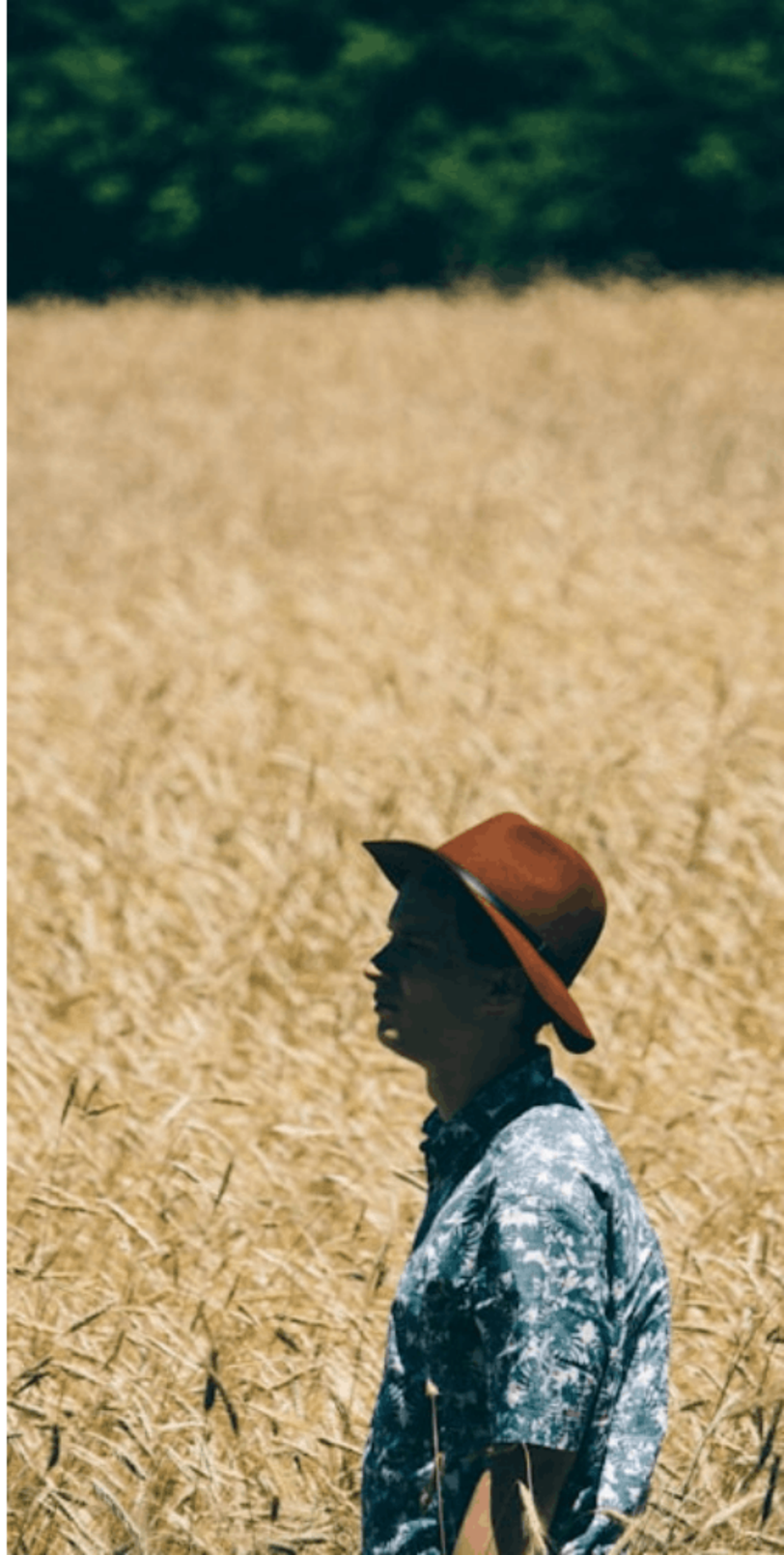
Automation and UI Excellence

- Founder of Batteries Included (CEO)
- Facebook (Ads Tech Leads, Dev Efficiency, Distributed Datastores)
- Cloudera (Apache HBase)
- Startups (Data, Stats, Frontend)
- Microsoft (C# and WPF)

Example Platform Goals

Always strive higher

- Help build sustainable & maintainable software for our users
- Speed up their time from idea to in production
- Maintain reliability of all services and help when there are issues



#1 Source of incidents?

It's me. Hi.

I'm the problem, It's me

Two Causes

Major root causes of issues, not the only

We didn't understand the problem

We didn't properly test or verify

Not Enough Testing

- We didn't have proper test coverage
- We didn't have integration tests for that
- We didn't test restoring backups
- We haven't tested failing over to redundant systems

Didn't Understand

- We didn't understand the tool
- We didn't understand success
- We didn't understand there was a more straightforward solution
- We haven't understood our systems at that load/size/scale

What's the cause

1

New Project Change

A developer adds a new DB user and increases storage for the production environment, mistakenly including the wrong units in the storage size.

2

Onto the pipeline

Testing locally works well, test environments pass, so the other engineer on the project approve the pull request.

3

Production Impact

In the best case, our gitops pipelines start failing now, stopping all other forward progress of other teams. In the worst case: pods don't schedule.

Everything went wrong

- Understanding there were 2 changes
- Understanding YAML schemas and input value types
- Testing changes for specific environments
- Testing signals being trusted when there are gaps
- Understanding who will code review
- Understanding how to continue deploying after failure

Make a tool for that

Automate the most frequent or scary changes that your users make

Create UI's that validate and ensure testing of a changed environment

Repeat, until almost all operational changes are through tools

Risk **vs** Reward

Bloudering: Small fall without safety nets

Climbing: Large safety net means controlled assents

Please don't climb free solo in production

https://commons.wikimedia.org/w/index.php?curid=127779918

UI Patterns

Not everything works for all places, but these patterns have worked well.

Modals/Pop up encourage speed

Users instinctively try and get off of pop ups or modals so they pay less mental attention to correctness

Early Input Validation

Structured input and early feedback make users more likely to follow or heed the advice and not get frustrated.

Event/Action attribtuion

Knowing what systems or person took action at what times is critical to finding and fixing production issues.

Hilight recent changes

Making recent changes easily discoverable allows faster debugging and encourages exploration, aiding in onboarding and reliability.

Batteries Included

Fair source, all inclusive software infrastructure platform

https://www.batteriesincl.com/

elliott@batteriesincl.com

https://github.com/batteries-included