From Chaos to Consistency

One custom DDEV add-on for optimizing DX across 100+ projects

Bill Seremetis (bserem)

DRUPAL DEVELOPER DAYS

annertech

Who Am I?

Bill Seremetis

- Tech Lead
  Annertech Managed Services
- Drupal developer
- DevOps Architect
- Helped companies transition to DDEV
- Active Drupal & DDEV contributor

drupal.org/u/bserem

Who Is Annertech?

Annertech

- Drupal agency
  Formed in 2008
  HQs in Ireland and the UK
- 47 people, ~30 developers
- 100+ active projects

www.annertech.com

1

Who is this talk for?

Is this for you?

Team Leaders & Agency Owners

Multi-Project Teams of Any Size

One-Man Armies with Many Projects

The principles also apply to teams working on a single project

2

The Challenge (Chaos)

The Scale

- How do you keep track of what / where / how?
- How do you make it consistent across projects?
- How do you ensure that lessons learned are applied to other projects?

The Chaos

01

Repetitive tasks

Good workflows exist, but they're manual and inconsistently applied

02

Knowledge silos

The right / efficient way to do things lives in a few people's heads

03

Onboarding friction

New developer productive time is lost for the wrong reasons

3

The Solution (Consistency)

An Add-on Is Whatever You Want It to Be

- Most think of add-ons as service providers:
  Solr, Redis, Cypress, Adminer…

- But in reality an add-on is a set of files:
  hooks + commands + scripts + config

- **It's a distribution mechanism**

Architecture: What's Inside

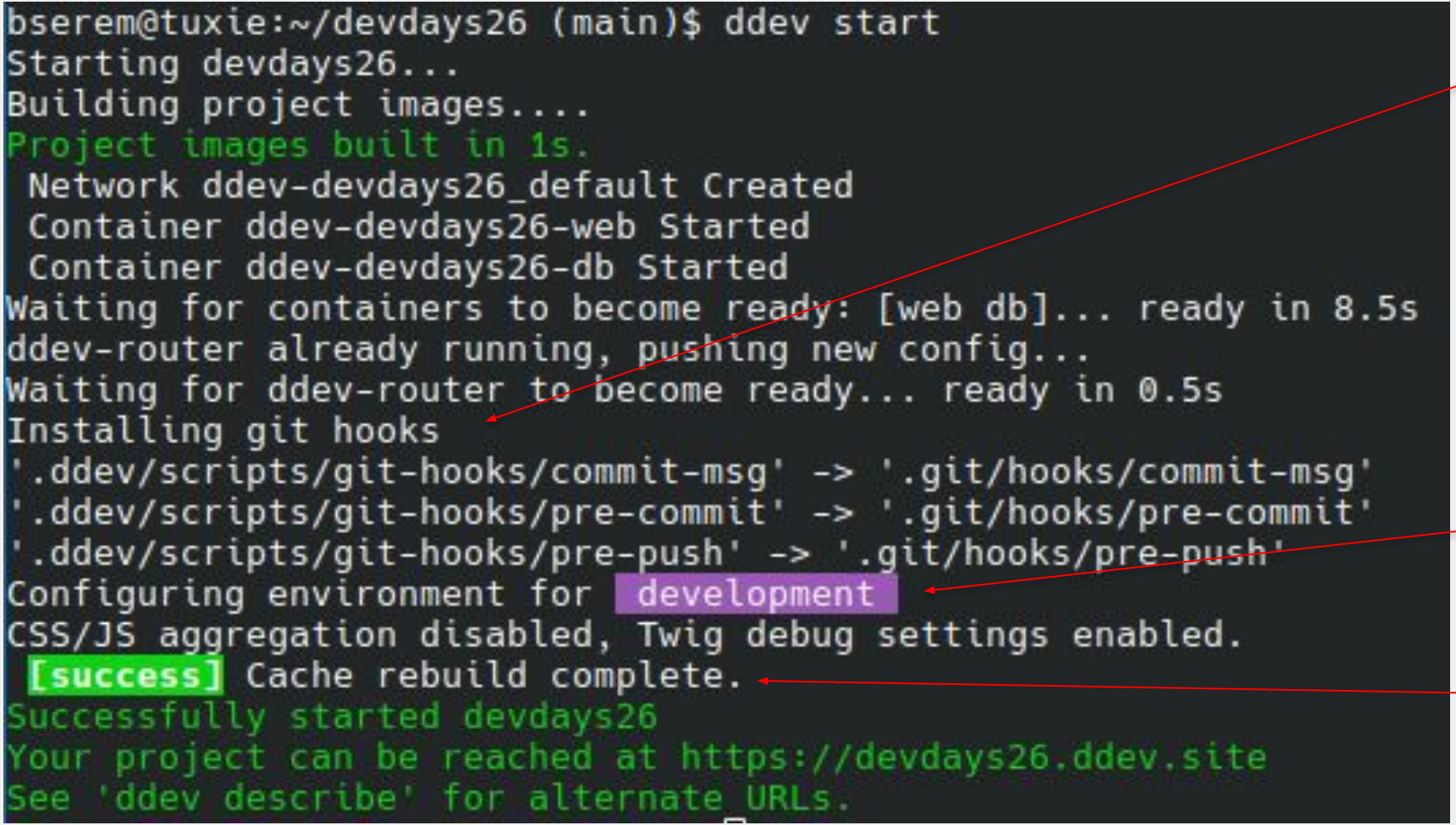
Custom commands
Your team's workflows and processes as first-class CLI tools.

DDEV hooks
Encode what should always happen at key checkpoints: project start → install git-hooks, db-import → sanitize, etc.

Boilerplate Configs, Scripts, AI Prompts etc.
Reusable logic distributed as part of the add-on

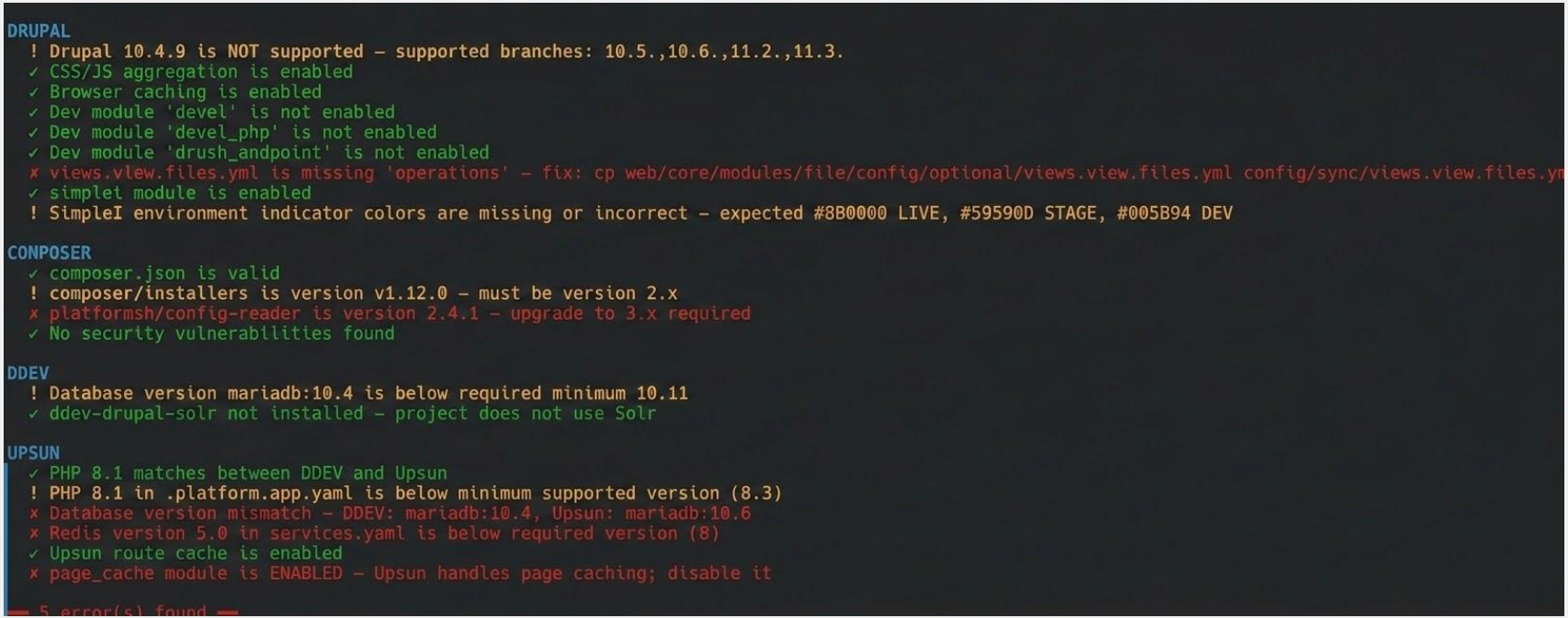
Per-project overrides
The escape hatch when a project needs something different

One update propagates improvements everywhere


bserem@tuxie:~/devdays26 (main)$ ddev start
Starting devdays26...
Building project images....
Project images built in 1s.
Network ddev-devdays26_default Created
Container ddev-devdays26-web Started
Container ddev-devdays26-db Started
Waiting for containers to become ready: [web db]... ready in 8.5s
ddev-router already running, pushing new config...
Waiting for ddev-router to become ready... ready in 0.5s
Installing git hooks
'.ddev/scripts/git-hooks/commit-msg' -> '.git/hooks/commit-msg'
'.ddev/scripts/git-hooks/pre-commit' -> '.git/hooks/pre-commit'
'.ddev/scripts/git-hooks/pre-push' -> '.git/hooks/pre-push'
Configuring environment for development
CSS/JS aggregation disabled, Twig debug settings enabled.
[success] Cache rebuild complete.
Successfully started devdays26
Your project can be reached at https://devdays26.ddev.site
See 'ddev describe' for alternate URLs.

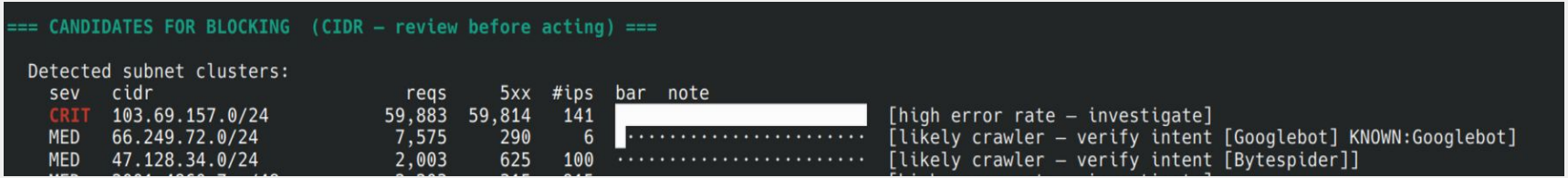
Encoding Institutional Knowledge

- Accumulated knowledge about what "correct" is for your team
  `ddev sanity-check:` one command audits Drupal, Composer, DDEV, CDN etc



Encoding Institutional Knowledge

- Quick access log forensics to help put out fires
  ddev loghound: Scans access.log in seconds and provides a quick analysis

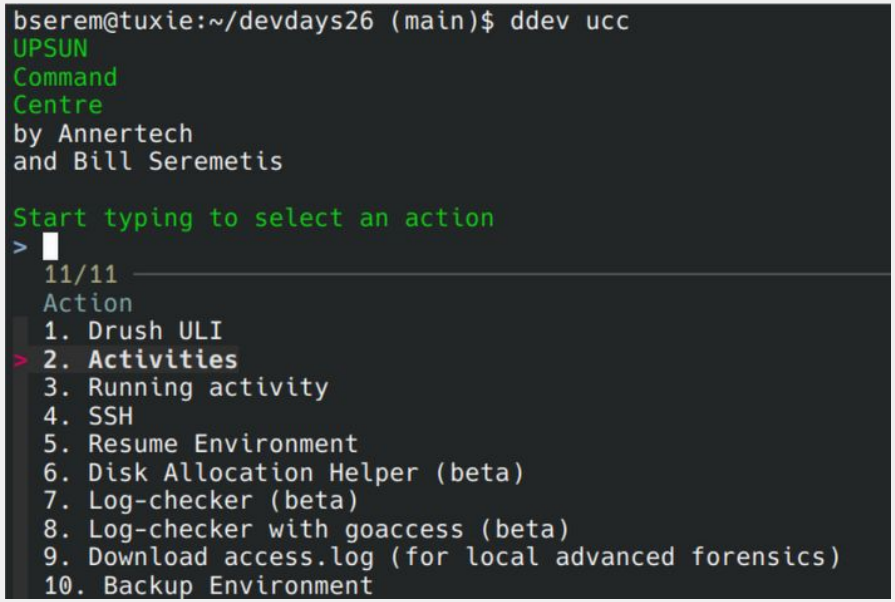


```
=== CANDIDATES FOR BLOCKING  (CIDR — review before acting) ===

  Detected subnet clusters:
    sev   cidr                 reqs     5xx  #ips  bar  note
    CRIT  103.69.157.0/24      59,883  59,814   141  ████████████████████  [high error rate — investigate]
    MED   66.249.72.0/24        7,575     290     6  ·····················  [likely crawler — verify intent [Googlebot] KNOWN:Googlebot]
    MED   47.128.34.0/24        2,003     625   100  ·····················  [likely crawler — verify intent [Bytespider]]
```

```
## Incident

- **Dominant error:** HTTP 500 (100,737 hits, 29.1% of traffic)
- **Window:** 2026-03-29T23:44:21+00:00 → 2026-03-30T07:21:47+00:00
- **Peak minute:** 2026-03-29T23:49:00+00:00 (2,428 req in that minute)
- **Top contributing IP:** 103.69.157.166 (473 hits — check further)

## Indicators

- **Largest IP cluster:** `103.69.157.0/24` — 141 IPs, 59,883 requests (possible botnet, CDN, or shared network — check further)
- **Other notable clusters:** 204 additional cluster(s) detected
- **Possible IP floods:** 52 IP(s) exceeded 60 req/min (top: `104.248.138.45` @ 997 rpm — check further)

## Suggested next steps

- Review the **Candidates for blocking** section — review before acting.
- Cross-reference flagged IPs against the 5xx deep dive and AbuseIPDB before blocking.
- This is an offline heuristic report; verify findings with live traffic data and WAF logs.

real    0m9,645s
```

Encoding Institutional Knowledge

- Quick access to most common host commands (Upsun TUI)
  ddev ucc



Encoding Project Management Workflow

- Your team has a PM tool, a branching convention, a time-logging process
- All of that can live in the add-on — bringing it to the terminal where developers already are
- ddev branch: creates a git branch with specific patterns, **tying** a branch to a ticket
  ddev timelog: adds a timelog entry on active ticket
  ddev comment: adds a comment on active ticket
  ddev mr: makes an MR in gitlab
  ddev jira-description: updates ticket description with link to MR and notes

Switch to browser
Log into PM tool
Navigate to ticket, **click** tiny log time icon
***wait** for ajax…*
populate time and description in popup
click save

vs

ddev timelog 15min "*fast*"
ddev comment "*I'm fast*"

Enforcing Quality Gates

Git hooks, MR/PR templates, linter configs — **distributed automatically** with the add-on

Developers don't configure these — they live inside the project repo and get updated everywhere

pre-commit

Custom checks before every commit

commit-msg

Enforce commit message format

pre-push

Final validation before push

Encoding AI-Assisted Workflows

- Your team's curated AI prompts, versioned and distributed like any other tool
- An interactive TUI for AI agent workflows — right where developers already are

The Non-Command Commands

Wrappers — reduce cognitive load:

ddev fetch-db → wraps ddev pull X --skip-files --yes

Discovery — tell you what's available:

ddev lints → What linters does this project have?

ddev tests → What test suites can I run here?

The add-on as a map, not just a toolbox.

4

Pros and Cons

the honest part

What Worked, What Didn't

✅ What Worked

Easier communication:
a few commands to remember and pass on

One place to improve:
update the add-on, everyone benefits

Consistent branch naming → consistent git log → linking changes to tickets:
git history on steroids

Colleagues started sending in improvements
Once they saw what's possible, they shared their solutions and nuances with everybody else

❌ What Didn't

Add-on Documentation that got ignored
Slack announcements that scroll away

Maintenance overhead:
the add-on itself needs care and documentation

Adoption:
Took a lot of time for people to onboard

Lessons Learned

1. The add-on sets defaults; projects own their overrides

2. Workarounds are feature requests in disguise

3. You are building a product for your colleagues — **their friction is a bug report**

Victory is not measurable ONLY in minutes

ddev timelog → 30+sec to 3sec, at least once per ticket

ddev compile-theme → 10min to 2min, once a month

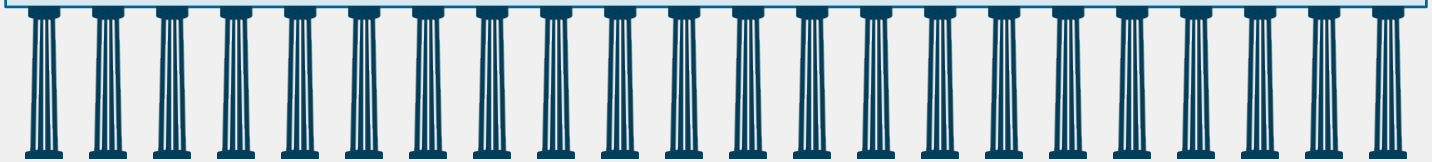
ddev ai-prompts → 60min to 2.5min, once per project/client

ddev sanity-check → witch hunting vs walk in the park

ddev upsun-controls → ~30min/day less context switching, every day

These are fine, but… nothing beats this:

> Whenever I pull down a project it reliably and consistently all works. I get a working site and a branch ready to go in seconds. It used to be the case that when I might have been switching context a lot I would have to spend half an hour a few times per week to resolve discrepancies to be able to start on a ticket.
>
> *Tony Paul Barker*
> *Developer at Annertech and LocalGov Drupal*



5

Building Your Own + Q&A

Start Here

Find your team's biggest pain

Encode it as one command

Ship it

- ddev add-on get ddev/ddev-addon-template
  - Use this as your base, build from there
  - Use hooks in config.team.yaml
  - Add custom commands
- Publish on GitHub, install with ddev get your-org/your-addon
  Use existing add-ons if they exist — don't clone them inside your add-on

TAKEAWAYS

1. DDEV add-ons are more powerful than most people realise
2. Standardization at scale requires both tooling and documentation
3. Start small: one shared command of a repetitive task is all you need

Thank You

Bill Seremetis

annertech

drupal.org/u/bserem

annertech.com

github.com/
Annertech/annertech-ddev