

Build faster than AEM

Experiences from enterprise projects

Phil Hawksworth, Netlify



hello

Phil Hawksworth

Developer Experience, Netlify

hello

@PhilHawksworth @indieweb.social

Developer Experience, Netlify

A little background

Developer Experience — Netlify

Group Technology Director — R/GA

Technical Director — The Team

British Telecom, Verisign, VM Ware, and various software houses

Building for enterprises

— working with enterprise infrastructures

Stories and experiences

A man in a dark suit, light blue shirt, and patterned tie is shouting into a black mobile phone held to his ear. His eyes are obscured by a dark redaction bar. The background is a blurred office setting. The word "Sensitivity" is written in large white font across the center of the image.

Sensitivity

Sensitivity

High profile decision and
expensive vendor selection

Large sunk costs
and invested efforts

Relationships
and dependency

3 stories

Unspoken costs, and the
power of **demonstrations**

Time constraints trumping
infrastructure policy

Efforts towards
evolving and **decoupling**

QUESTION

When working with AEM

what are your biggest challenges?

QUESTION

When selecting AEM

what were the biggest motivators?

STORY ONE

A short project made long

Unspoken costs, and the
power of **demonstrations**

STORY ONE

A brand site for a large enterprise

Product pages, editorial content

Localised for several local markets

Desire for a premium, luxury aesthetic

Adopt new company-wide components

Performance and device support issues

Multiple agencies had tried and failed

Design and F/E specialists to work with SI and platform team

Project time approaching 2 years

CMS complexity required developers

Developers had web dev, not AEM expertise (and vice-versa)

this can be simpler

STORY ONE

A proof of concept for measurement purposes

All page templates built and sample populated using a static site generator

1 developer
9 days build
1 day measurements

Not integrated into CMS, but proving capability and effort for f/e delivery

Side-by-side video of page load performance showing improvement of ~10 seconds

What about?

Deployment time?

Localisation?

Personalisation?

Interactive content?

Content management?

QUESTION

What else?

what other challenges might we raise here?

STORY TWO

A sprint to the finish

Time constraints trumping
infrastructure policy

STORY TWO

A campaign site for a large global brand

Few pages, with several campaign phases

One language and no content management

Campaign site retired shortly after event

Short lead time: 6 weeks to event

Must use global delivery platform

Lead time for dev environment: 4 weeks

Audit of configured dev environment before approval of QA, Staging and Prod



Time was the forcing function

STORY TWO

A stop-gap development environment which was production ready

While formal environment being provisioned, the dev team “prototyped” on Netlify

Ready-built campaign phases with instant releases solved

Stakeholder reviews on feature branches

Formal environments were not delivered and configured before the go-live date



It went wonderfully

QUESTION

Provisioning and approval

Is this ever a challenge? Techniques for expediting?

STORY THREE

Proving a concept

Efforts towards evolving
and **decoupling**

STORY TWO

Very large enterprise
and global brand

Eagerness to prove
composable and
Netlify workflow

Desire for additive
approach to AEM

Small team for a short
spike on a brand site

Platform team,
developer from SI,
product owner

Expose existing site
content from AEM as
content APIs

Present results to
platform stakeholder

Almost

AEM → JSON API

AEM → JSON API

“Possible”

AEM → JSON API

“Possible”

Content modelling
impacts this

QUESTION

Explored going headless?

Tips for AEM publishing content APIs?

FINALLY

Some lessons

from these experiences

Proving multiple concepts

Proof of concepts
to allow
measurements and
comparison

Demonstrate
workflows and
values in less
obvious areas

Inertia and pushback

Deep investment
brings deep
commitment —

Expect resistance.
Stay pragmatic.

Current challenges
might surface as
whataboutism —

Seek out existing
solutions or the
absence of them

It's not all or nothing

Avoid big bang and big risk — take an additive approach

Migration can be done gradually with Netlify redirects and proxying

[Video demonstration](#)

Select pilot projects with well assessed risk

Thank you

Let's have a chat

Phil Hawksworth, Netlify

