Growing Software Engineering Talent In-House at the Ordnance Survey

Welcome!

Our session will begin at 1900

Hi! I'm Teresa.

Senior Software Engineer

Joined the OS as a trainee 6 years ago

Comp Sci

Growing Software Engineering Talent In-House at the Ordnance Survey

via the

Trainee Scheme

"Start with Why"

Getting good software engineers

Getting well-rounded people

Getting our culture and values

New beginnings

Play the long game

Recruit for aptitude and enthusiasm over experience

Grow software engineers on the job

Ownership of
Learning

"Assault Course"
Teamwork

Passion for
Technology

Reflection

This post specifically addresses the Engineering Academy trainee recruitment, but the contents are applicable (albeit less directed) to the direct-to-team trainee experience.

What we're looking for

When interviewing for Engineering Academy trainees, we look for candidates to show us:

- That they can work effectively in a collaborative team
- That they can own their own learning, and learn independently
- That they have a passion for the kind of software engineering we do

Reflecting on Reflection

Whilst writing this post, I came to realise that there is a common theme running through everything I've written. That theme is "reflection" or "reflective practice", and it's worth elaborating on it here so you can see it more clearly in the detailed sections before.

The Engineering Academy interview panel aligns to the idea "We can teach technology, we can't teach behaviours". The three things we look for in interviews

Recruit for Aptitude over Experience

Getting well-rounded people

Life Experience

Behaviours

Diversity

Future Senior Engineers

Real Work

Intentional Recruitment

Important not urgent

Support

Time

Targeted teaching interventions

Engineer

Timelines

3 months

10 months

14-16 months

Retention

Before Pandemic

Post Pandemic

Graduate Scheme, Apprenticeship Scheme

We have both

Grads

Apprentices

Summing up

Growing Talent is a long game

but we find it's worth it

Teresa Binks

teresa.binks@os.uk

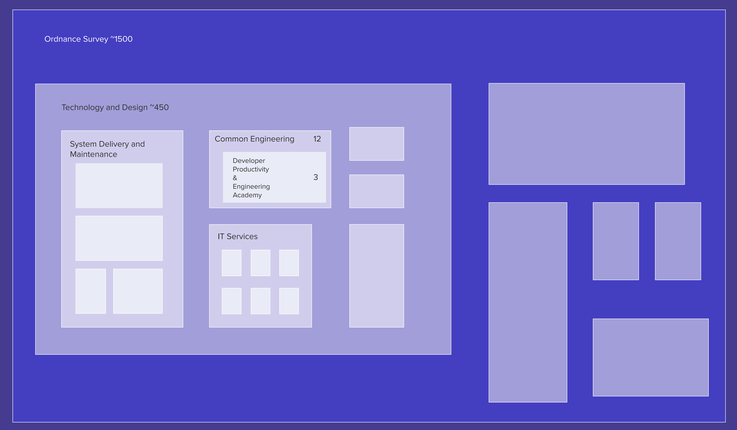
ordnancesurvey.co.uk

What questions do you have?

Routes

Direct to Team

Engineering Academy



Grow Software Engineers on the job

Key parts:

A backlog of important and not-urgent work

Support

Directed, specific teaching interventions

A focus on quality

Programming

C#, PowerShell

OOP, Hexagonal Architecture, TDD, ASP.Net

CI/CD/CD

Deployment pipelines

YAML in Azure DevOps

Values and Attitudes

Automation First

Cloud Native

Psychological Safety

Obligation to Dissent

Strong opinions weakly held

"Clean" Examples

*"Learners don't need **simple** examples, they need **clean** examples"*

With a narrative

Effective, not perfect

Azure for Normal People

The Pizzeria

Azure for Normal People

The cloud, without the cloud

The Pizzeria

Start with a console project	Manual testing
Order a cheese, tomato, ham, pineapple pizza	Look at the state of the pizza
Order a Hawaiian or Margharita pizza	If statements ➡ Switch case
Automate those tests	Test framework
Introduce an Oven	Dependency injection
Introduce a faster Oven	Test Doubles: Stub
Introduce a "Order Up" bell	Test Doubles: Spy
Hire a chef	Separation of Concerns
Teresa breaks the code	Triangulation
"Deliver" the pizza via Bike	Writing to 'FileVille'
"Deliver" the pizza via Car	Inheritance, writing to "TableHampton"
"Deliver" the pizza via Car	Interfaces, writing to "RESTminster"
Refactoring, seeing patterns	mocq Hexagonal Architecture - Ports and Adapters