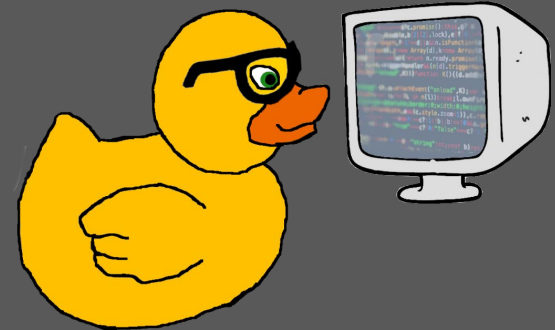


How we introduced

Quality Engineering

and improved everything

A technology leaders guide to iterative improvement



Who am I?



Stuart Thomas

Balding, middle-aged, white guy

Engineering leader

Publish my ramblings at TheQualityDuck.co.uk

I talk and write about all things Quality, Testing, DevEx, and Leadership

I'm here to tell two stories...

How we introduced Quality Engineering and improved everything

My time at Zoopla, and how a quality transformation transformed how we built and delivered software.

The learnings that I carried forward

My time at Humara, and the learnings that I will continue to take forward.

480

How we reduced deployment lead time by

HOURS

Where we started

- 100% manual testing
- Fortnightly manual deployments
- System offline for hours during a deployment
- Multiple Production instances deployed days apart

Low confidence, High risk

DORA:

- Deployment frequency: 2 weeks
- Deployment lead time: 3 weeks
- Mean Time To Recover: Days
- Change failure rate close to 100%

Taking the first step

Introducing Automated Testing

Start Small

- Pick technology that works for the whole team
- Start with small quick to write tests
- Focus on scenarios that have immediate value
- Reduce dependency on manual tasks

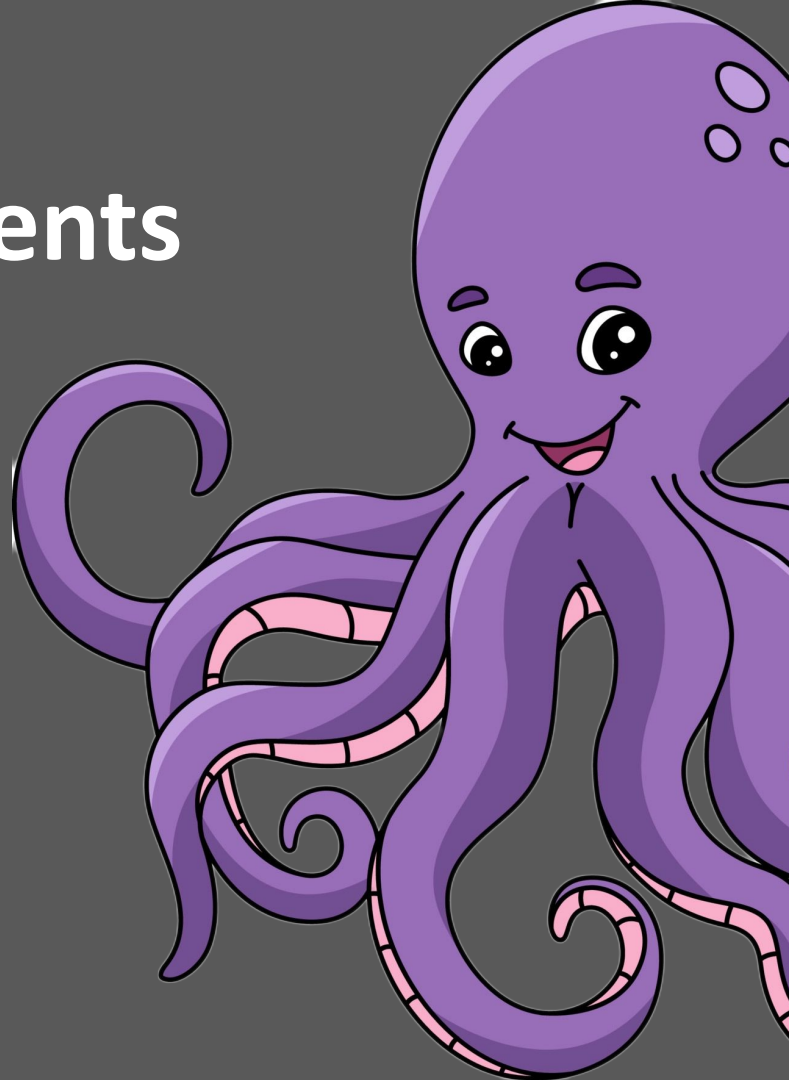
Continue to build high value tests

- Reduce dependency on specialist knowledge
- Automate tests that:
 - take a long time to complete manually
 - are prone to human error
 - people don't like to do



Unlocking Deployments

- The tool choice doesn't matter
- Use tools to standardise your deployments.
- Open up the process to broader review
- Empower people to make improvements
- Make it easier to experiment



Shift **Left**

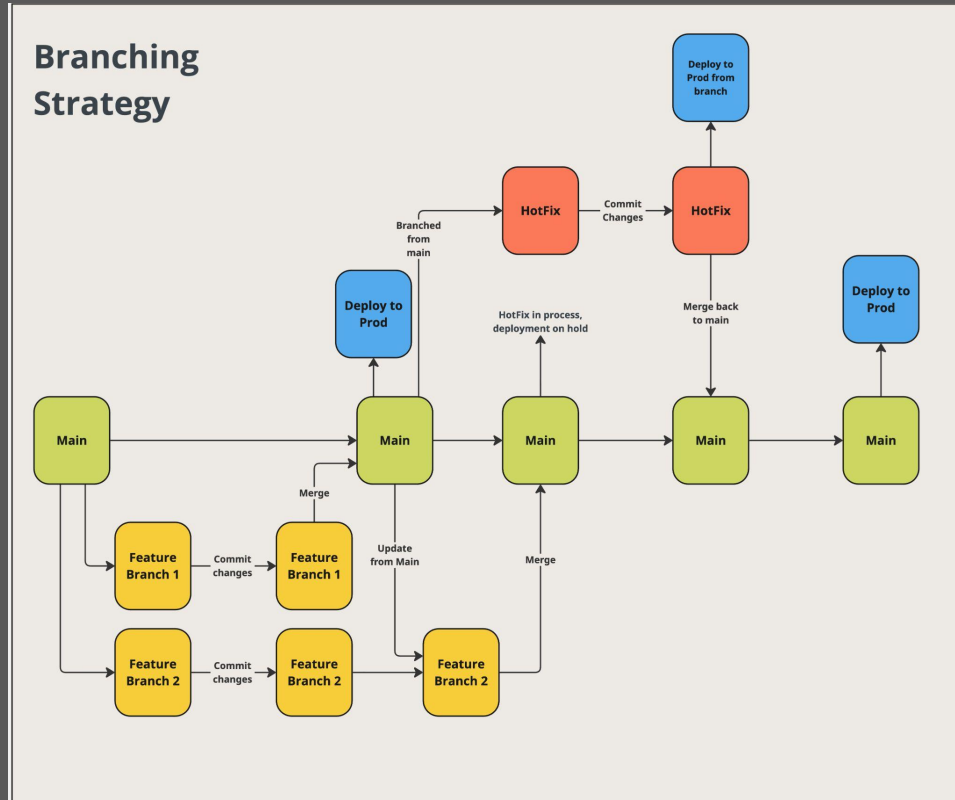
Collaboration

- **Everyone can test:** Quality, Software, Data, and Product all contribute to testing throughout the SDLC.
- **Reduce Bottlenecks:** Eliminate the “over the fence” culture
- **Define “Good” upfront:** Write tests before code and well defined suitable acceptance criteria / checks.
- Use **Red / Green** testing techniques
- Results in more time for **exploratory testing**



Branching

- **Always Deployable:** Enable Frequent Delivery
- **Short-Lived Branches:** Reduces merge conflicts and delivers value more quickly
- **Feature Toggles:** Delivers changes frequently and safely.
- **Increased Confidence:** Small changes tested and deployed often
- **Fast Feedback:** Less context switching and faster iteration
- **Deliver more value** more often with smaller changes



Ephemeral Test Environments

- Test environments on **demand**
- **Single use** – Build, deploy, test, destroy
- Enables **always deployable** main branch
- **Isolate work** from other changes
- Encourages independently **deployable** and **testable** changes
- Enables **fast feedback** to engineers
- **Reduces** cognitive load



Build trust in your process

Having confidence in the reliability of your tools and tests is key!

Not rushing to do all the deployments all of the time helped teams to build trust in the reliability of tests and deployment tools.

We achieved this through “manual” deployments using our automated tools, so that we were still on hand should there be a problem. This gave us the confidence to increase frequency of deployments and once we were confident the process worked, we could move on to the next step.

Automate All The Things!

Fully automated deployments with alerting

- Automatic release creation twice a day
- Deployment to a staging environment
- Automated regression testing
- Automatic promotion to production
- Alerting of issues with deployments



Where we started

- 100% manual testing
- Fortnightly manual deployments
- System offline for hours during a deployment
- Multiple Production instances deployed days apart

Low confidence, High risk

DORA:

- Deployment frequency: 2 weeks
- Deployment lead time: 3 weeks
- Mean Time To Recover: Days
- Change failure rate close to 100%

Where we got to

- Continuous deployments
- Automated regression testing
- Fully automated deployment pipeline
- System remained online during deployments

High confidence, Low Risk

DORA:

- Deployment frequency: **Multiple times per day**
- Deployment lead time: **Hours**
- Mean Time To Recover: **Minutes**
- Change failure rate: **< 5%**

What we learnt

- Not everything works first time – that's ok!
- You **will** encounter issues no one predicted.
- You will adapt and overcome.
- DevEx and Trust are **essential**.

Experiment, learn, react



The learnings carried forward

Where we started

Humara - Guided Selling Engine

- Three weekly deployment cadence
- Regression testing took a week for 5 people - Test automation existed but was low quality
- **High change failure rate, low confidence releases, slow release cadence**

Humara - Virtual Sales Agent

- Greenfield opportunity.
- Lacked process
- Slightly wild west, but a great opportunity to get it right early.
- **A chance to shape good from the start**

The learnings carried forward

What we achieved

Humara – Guided Selling Engine

- Modernised test automation – created new tests with lower coverage, but higher confidence.
- Improved deployment tooling – made it easier and faster to deploy changes
- **Achieved daily deployments, lower change failure rate, and faster recovery times.**

Humara – Virtual Sales Agent

- Built a development process around CI/CD and shared understanding from the start.
- Used ephemeral environments to shift all types of testing as early as possible.
- Created fast feedback loops that provide early confidence.
- **Deploy multiple times per day, almost zero change failure rate.**

Two things are consistent

Quality
Engineers
make things
better for
EVERYONE.

Teams
should move
fast, and
~~break~~ **LEARN**
things!

Lets connect

Questions?

Come see
me for office
hours

Scan for
socials

