humara

How we introduced Quality Engineering and improved everything

Brighton | UK

Who am I?

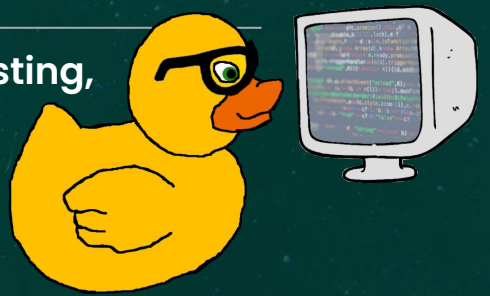
Stuart Thomas

Balding, middle-aged, white guy

Head of Quality at Humara

Publish my ramblings at TheQualityDuck.co.uk

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



We didn’t know what we were doing…

- Quality Engineering wasn’t the goal
- All initiatives were led by Quality
- DORA:
  - Deployment frequency: 2 weeks
  - Deployment lead time: 3 weeks
  - Mean Time To Recover: Days
  - Change failure rate close to 100%

480

How we reduced deployment lead time by

HOURS

ZOOPLA

humara

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**

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

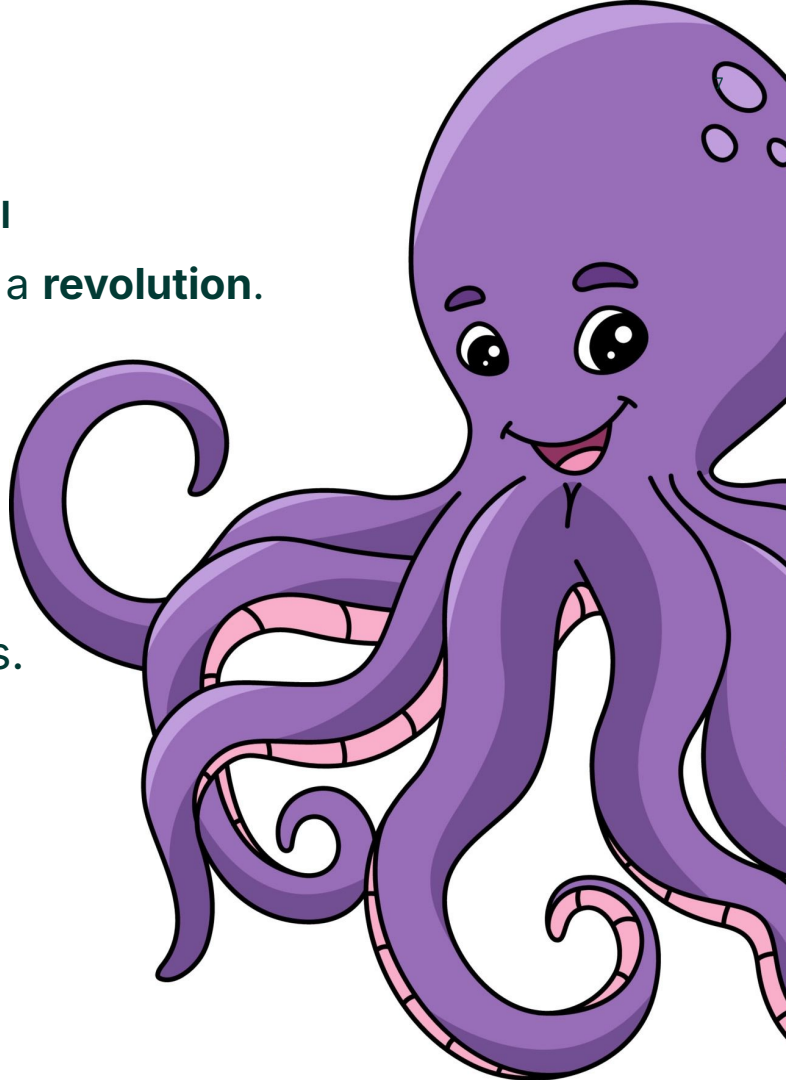
- Reduce dependency on specialist knowledge
- Automate tests that:
  - take a long time to complete manually
  - are prone to human error
  - people don’t like to do

Unlocking Deployments

Using Octopus Deploy made deployments accessible to all

The introduction of Octopus Deploy was the start of a **revolution**. We went from only a select few having access to complete production deployments to empowering the **whole of our engineering team** to be able to deploy.

- Octopus Deploy standardised our deployments.
- Opened up the process to broader review
- Empowered people to make improvements
- Made it easier to experiment



Shifting Left

Break down the fence: Collaborative Quality Ownership

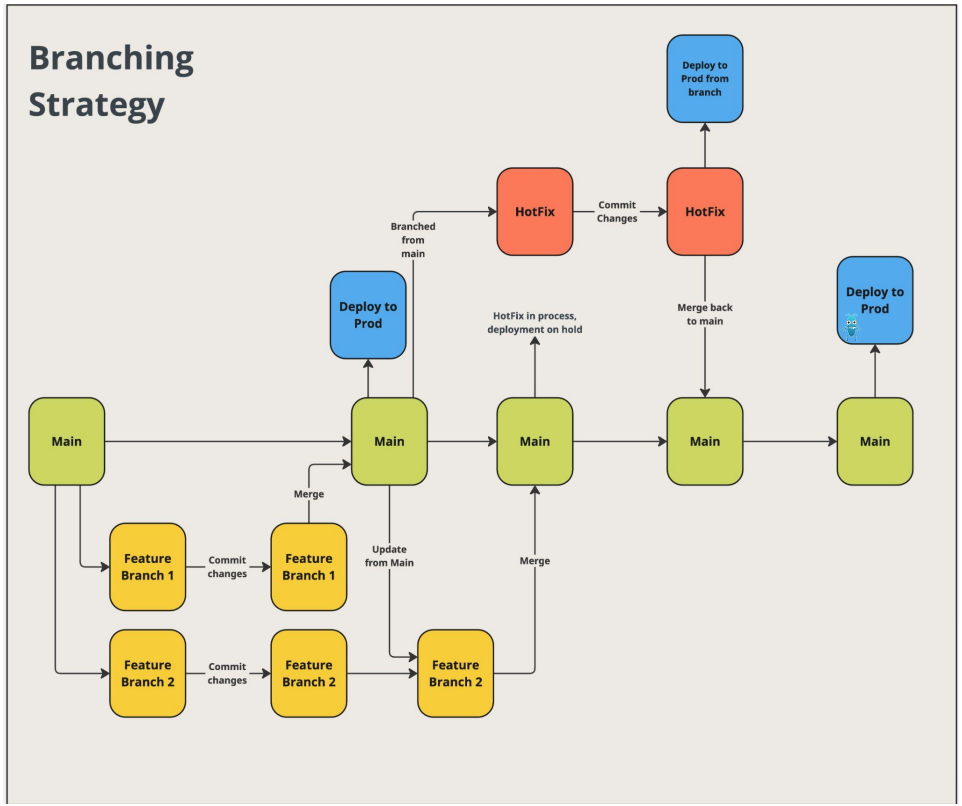
- **Everyone can test:** Quality, Software, Data, and Product all contribute to testing throughout the SDLC.
- **Reduce Bottlenecks:** Eliminate the “over the fence” culture
- **Define “Good” upfront:** Write tests before code and well defined suitable acceptance criteria / checks.
- Use **Red / Green**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

Shifting Left

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

humara



Ephemeral Test Environments

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

Likelihood x Severity = RISK

- Create **shared** understanding and language around risk
- Unify understanding across the **entire** business
- Have centrally defined **measures** for likelihood and severity
- Plan for **failure**
- Balance **Risk** vs **Reward** vs **Time To Resolve**

humara

5x5 RISK MATRIX

LIKELIHOOD ↓ / SEVERITY →	1	2	3	4	5
1	LOW 1	LOW 2	LOW 3	MEDIUM 4	MEDIUM 5
2	LOW 2	MEDIUM 4	MEDIUM 6	HIGH 8	HIGH 10
3	LOW 3	MEDIUM 6	HIGH 9	HIGH 12	EXTREME 15
4	MEDIUM 4	HIGH 8	HIGH 12	HIGH 16	EXTREME 20
5	MEDIUM 5	HIGH 10	EXTREME 15	EXTREME 20	EXTREME 25

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**

Where we got to

- Continuous deployments
- Automated regression testing
- Fully automated deployment pipeline
- System remained online during deployments
- **High confidence, Low Risk**

What we learnt on the way

- Not everything works first time - that's ok!
- You will run into issues no one predicted, you can adapt and overcome
- DevEx and Trust are essential!
- **Experiment, learn, react.**

humara

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**

Where we got to

- Continuous deployments
- Automated regression testing
- Fully automated deployment pipeline
- System remained online during deployments
- **High confidence, Low Risk**

What we learnt on the way

- Not everything works first time - that's ok!
- You will run into issues no one predicted, you can adapt and overcome
- DevEx and Trust are essential!
- **Experiment, learn, react.**

We didn't do it overnight...

October 2019

Fortnightly deployments

Lead time from first commit of 3 weeks

Test Automation

Automated most repeated tests - Release smoke tests and Regression tests

Unlock deployments

Using Octopus Deploy made the deployment process accessible to more of the team

March 2022

Build confidence

Use the new tools to deploy more frequently and gradually build confidence in them

Automate the pipeline

Automated the remaining manual deployment steps

How did it impact DORA?

humara

- DF: Multiple times per day
- DLT: Hours
- MTTR: Minutes
- CFR: <10%

How has it gone at Humara?

Quality Engineering at Humara

- Ephemeral environments allow teams to build and test **quickly**
- Automate away **repetitive** manual work
- Take a **holistic** approach to testing
- Build a common language around **risk**
- Improving **DevEx** improves **Quality**
- Take time to develop **trust** in changes

Quality Engineers make things better for EVERYONE.

Move fast, and ~~break~~ LEARN things!


A Brief History of Time
Questions?