

Intro to DevOps

Course Workbook

To be completed by you and your team



Course Expectations

What I currently know about DevOps:

What I would like to learn on today's course:

Introduction

The content in this course is designed to help you start the process of moving towards a DevOps culture. It scratches the surface of many of the areas you'll need to begin to understand in order to start to work together as one unit. DevOps is a journey, one we may be starting today - it is not a journey which concludes at the end of the training; therefore you are encouraged to continue to have the discussion online, through colleagues at the School of Tech and through your own learning. We are here to help you in your journey as we travel along our own.

Please feel free to make photocopies of the workbook section to help you and your team evaluate your current working practices and where your strengths and areas for growth are.

Your Company:

List the teams you have - e.g. R&D , Development, Operations, Management:

Your *Own* Team Members:

Your team's overall key Strength:

"Most Important" decision maker:

/ :
/ :
/ :
/ :
/ :
/ :
/ :
/ :
/ :
/ :
/ :

"Least Important" decision maker:

Key Strengths:

Key strengths of individuals in your team (list as many as appropriate):

Key areas of strength in your company:

e.g. Good communication / Quick turnaround of ideas and products

Key areas to focus on across your company:

e.g. Increase team communication / Need to fail faster and learn

Timescale you'd like for changes in an ideal world:

Current infrastructure costs:

(cost of servers you have running / software production etc.)

Budget you have for implementing changes:

Company Organisation Chart:

Draw a company organisation chart ensuring that you:

- Label the areas that communication bottlenecks appear.
- Label the areas that cause a block to production of your product or service.
- Label the areas of strength and reasons why.
- Label the areas you yourself can impact.

Describing your company

Hardware stack:

Software required:

Development Cycle Length:

e.g. 1 week / 1 month / 3 months / 6 months

Product Release Timeframe:

e.g. monthly / quarterly / bi-annually / annually / once in a blue moon

Timescales and Pipeline considerations:

Team considerations:

Thinking about Quality Assurance, when do the QA team get involved and how long do they have to test the product?

Thinking about the development cycle - who sets the task list and decides on how long tasks take?

Thinking about the operations / release cycle - when do the Ops team get involved in the process and how long / resources do they get?

Draw your system development cycle providing names of individuals (if possible) for each of the teams:

Company Processes

Design and Scoping

- How are features decided?

- What is your client feature request process?

- What is your Redesign Process?

- How do developers and clients feedback on designs?

Software Development

- Creating a new feature - how do you scope the feature and decide it will be built?

- Patching an old feature - what is the process for requesting a patch?

- Upgrading a feature - what is the process for deciding to upgrade a feature?

- Retiring features

Quality Assurance

- Testing new features - do you test new features? If so, how?
- End to end testing - do you test at the code level & the interface level? How?
- Finding and reporting faults - What is your fault reporting method? What happens when you find a fault?

Deployment

- Setting up production environments - Do you run your production environment? If so, how is this scoped-out?
- Applying software onto systems - How do you deploy your software? Who is responsible?
- Scaling - Do you scale? If so, how do you achieve this? What type of scaling methodology do you use? Why?
- Responding to failures - How do you respond to production environment issues?