

Working with developers - intervention without overstepping

Eleanor Mollett

I was once a
delivery
manager
worried about
overstepping



Often we spend our time ...

- Chasing updates
- Updating jira
- Updating delivery charts
- Being the shit umbrella
- Doing the admin
- Coaching from the
sidelines

But how do we help?

How much should we understand what our devs are doing?

How much should we intervene in what our devs are doing?

How can we actively help with delivery?

Step away

Technology choices

Tread carefully

Dev practices

Get stuck in!

Dev processes

Step away

Technology choices

- Architecture
- Languages
- Frameworks
- Dependencies
- Databases

Tread carefully

Dev practices

- Pairing
- Code reviews
- Refactoring
- Automation
- Working in the open
- Collective ownership
- Coding standards
- Test driven development

How do I influence good practice?

- Learn about them!
- Find out what's happening
- Find out what your devs want to improve
- Vocally support good practice
- Make it clear the time is available
- Make expectations clear

Pairing

— — —

Good

- making time available
- supporting against managers
- being clear its expected

Bad

- asking about it every time
- forcing unwanted pairs



Get stuck in!

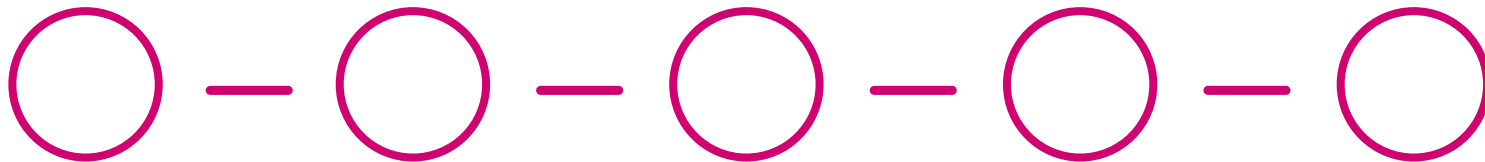
Dev processes

- The deployment process
- Managing tech debt
- Managing bugs
- Spikes
- The tech backlog
- When is done done
- Involvement in design and research

How do I influence dev processes?

- Internal process improvement
- Focus on quality + reducing rework
- Roadmaps
- Prioritisation
- Dedicated % of sprint/prioritised work

Mapping out pain points in your release process



Build

Test

Deploy to pre-prod

More tests

Deploy to production

Long build times

Long running tests

Access to pre prod environments

As before plus -

Change release process

Bad version control

Flaky tests

Alignment of pre prod with live

Not running against same environment

Change release board

Bad branching process

Commented out tests

Access to do deployment

Release windows

Low quality tests

Manual steps

A team other than the developers

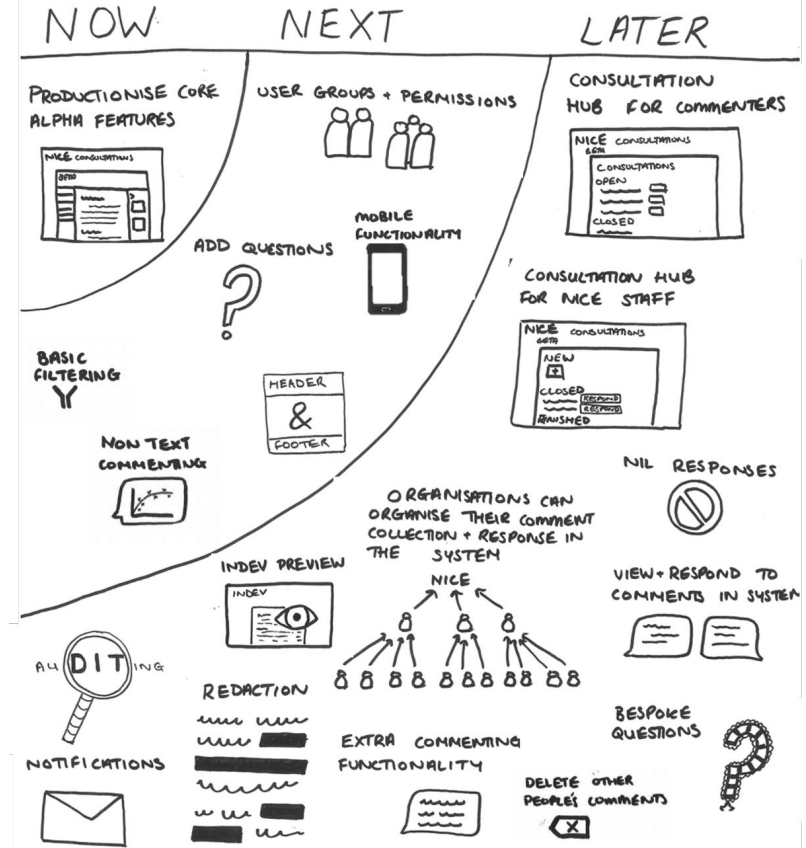
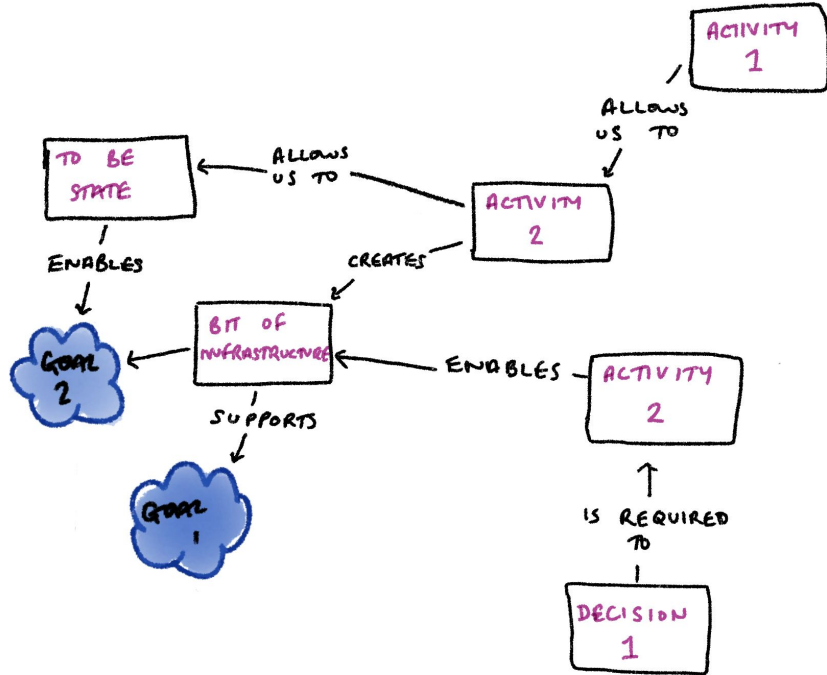
Technical backlog & prioritisation

Help put a framework together for prioritisation, and work through the backlog with them

Urgency v importance is a good starting point



Technical roadmaps



Should devs also be doing delivery stuff?

Yes!

- Leading standups
- Running retros
- Presenting at showcases
- Prioritising the backlog
- Writing stories
- Managing their own stories
- Talking to stakeholders

Remember!

**It's you and the developers vs the
deadline,**

**not you and the deadline vs the
developers**

Thanks!

Any questions?