Norfolk Developers would like to explore the stacks that drive our fine county. We'd like to explore your process and how you go from A to B.

In short, we're looking at how you:

- Analyse business/user requirements & translate those into user stories (or similar)
- Project manage the delivery of those requirements
- Deliver them
- And, Test/QA & gather feedback

The format for this will be a 15min lightning talk, with this document guiding you on some talking points, but, by no means sets an expectation. The questions and/or topics are not requirements to partake, or a metric by which to measure, and instead are meant to jostle the brain cells.

We will invite 2-3 organisations (or makers) to join us and talk us through their process and bring them together at the end as a panel for Q&A.

These events will (can) also tie directly into a directory we will establish on the new https://www.norfolkdevelopers.com/ website, where we will list your organisation and the stack you use.

Talking Points

- ### What tools and/or methods do you use to gather user requirements?

  Do you have "user groups"?
  Regular meetings with customers?
  Accept enhancement requests via GitHub, or with a product like Uservoice?
  Dogfooding?

- ### What tools and/or methods do you use to translate user requirements into actions or user stories? And what methods do you use to prioritize these?

  Regular meetings with account managers, support teams, or tech teams to explore user requirements, and or their own requests?

- ### What tools and/or methods do you use to manage the delivery?

  Are you using PM tools (e.g. Jira, Trello etc)?
  Methodologies like:- Agile, Scrum, Shape Up?
  What aspects of the methodology do you find work best, and, perhaps what do you think is unique about your own version?
  Is your team remote? Office based? Are you using standups?

- ### How are you delivering?

  What's your tech stack? (from back to front, or from schematic to consumer hardware)
  What's your local dev environment look like?
  Are you using a UAT or staging environment?
  VMs, Docker, Kubernetes?
  Monorepo? Private package repositories?
  Continuous Integration, Continuous deployment?
  What's your strategy for deployments (release branches, tagging, artifacts?)

- ### How are you testing, gathering QAs, or user feedback

  This may be answered in part within the first question if relevant.
  Do you operate unit, feature, e2e, or integration tests?
  Live Chat for Customers?
  Metrics (New Relic etc)