Session: 2
Room: Committee Room 2

Session title : Is Agile 'fast enough' anymore

Session leader : Veronica

Volunteer to continue conversation after :

Notes taken by : Prateek Buch, Policy Lab

Notes

Merged with session on doing procurement better

Are Agile multidisciplinary teams being seen as the new waterfall teams, in light of AI seemingly doing some tasks faster? Does that resonate? What could be done if so?

At a govt dept, lots of forward-deployed engineering, and lots of senior leadership ask 'why does it take so long to do A, B or C?' which are questions agile teams used to ask of waterfall teams!

On another note, some discovery processes can take several years, so sometimes questions around speed can be valid!

Post-its on a grid

- Should we reframe to ask 'in an AI world, are people seen as too slow?' People run agile processes, so is there a time pressure now because of AI seemingly being faster?
- Agile can be a tick-box, what if it isn't that and does it mean to be agile?

- Fast enough at what? Now or next year? What about technical debt, especially when you don't pay attention to certain practices within agile manifesto such as technical or continuous improvement?
- Big A Agile and little a agile - some org, same project, a team might do agile on a word doc and get to a word doc, compared to a team with lots of Agile tools/processes, slowed things down
- Isn't agility an emergent property such as better outcomes, not a thing such as speed, in itself? Agile teams use AI tools to achieve that, and need to define what good looks like - sometimes a greater volume of code might be a good output/come, but there may be harder to measure things that meet 'what good looks like'
- What sort of methods is an agile team using - is it optimised for research/exploration, lean use of resources, etc? Spending time on eg user research on a deterministic area, maybe they aren't being fast enough? And what about the context the team is working in, what are the cultures and friction points the team is operating in? Show and tells are common, but how do we show what we've achieved when practising agile - we're asked what we changed, and to prove it with hard numbers
- Can get stuck in procedures of Agile ("where's your Kanban board?!", without adapting to context to meet objectives. Putting constraints on eg length of exploration (six weeks) allows you to move onto delivery.
- Need to be senior enough or close to decision-making to influence, when we're asked "what if we inserted Tool X would it make things faster?" That could be helpful, but it's also helpful to ask whether leaders want speed, innovation, better outcomes? Trade-offs must be discussed throughout an org
- Working in an agile way can be seen as risk management through faster feedback, good argument against just banging out code
- From user research perspective, don't use the word Agile any more - as it leads to commissioners saying they don't get what they want. Don't even use the phrase MVP, as it is seen as not doing the full job that's needed - when priorities shift, things discovered during agile research gets lost
- Stakeholders love to see their teams as agile but they still have waterfall/gantt milestones - need to empower agile teams to say 'sorry your feature might not happen just yet.' Early conversation with leaders on whether they prioritise exploration, speed, product delivery, meeting a political need… that helps resolve tension between milestones and being user-centred
- Do we have organisational memory of how bad software delivery can be without agile? Been going for 25 years so has had some impact!
- Agile within policy means that teams can only work as fast as the trust that leaders place on teams. If AI tooling can speed up say prototyping, that can create time to build trust, as it creates space to reveal what users needs and what solutions can look like. Must make sure tooling doesn't do the opposite and diminish trust
- Often get scope creep - agile used as a way to hide scope creep. If there's trust in an agile team, can help hedge against a top-down ask from say a Minister - gantt charts can be seen as a way to control deliverables, making it hard to be agile and pivot
- Agile is meant to be a team thing, often forgotten that teams are problem solving units rather than plucking people from here there - so squads end of storming norming from scratch.
- Team-based barriers, including stakeholders wanting different things