White paper: delivering with postagile

By Mark McKee and David Wilks, 23 May 2025

Abstract

Business leaders have become increasingly dissatisfied with what can seem like a lack of commitment to technology delivery. They (quite reasonably) want to get target delivery dates and are not interested in story points or anything that smacks of vagueness. They wonder if they are getting value for money from scrum masters and product owners. 47% of agile transformations fail and 80% of software features are barely used. The *17th State of Agile Report* concludes that agile is having difficulty keeping up.

Nobody argues with the agile principle, "Working software is the primary measure of progress". Yet, this core idea seems to get lost in over-engineered frameworks, complex estimation techniques, and roles that don't seem to add value if you have competent engineers who understand the business domain. So our problem now is that agile, and in particular, agile transformations, are not delivering value from the perspective of business owners and leaders who ultimately want to save money and speed up delivery.

Background

Building software products is complex because creating something new entails uncertainty. Will the customer use this new product? Will it generate the revenues we hoped for? Will it beat the competition? Prioritising carefully with our limited capital and human resources is vital to success. Pick the wrong things and you just sink time and money for no real competitive advantage.

There is always a list of problems for a business to solve, and most of them relate to giving customers a better experience that will grow your business. Even when you identify the right

priorities to build your business, leaders get frustrated if product delivery teams don't commit to reasonable delivery timelines. Pithy user stories, sprint retrospectives and story points don't cut it when leaders need to know the time to money. We need to call out what isn't working in the world of agile.

Solution

Wouldn't it be better if teams made strong commitments to delivery timelines with real confidence that what they're bringing to the market will make customers love your products? This is where *postagile* comes in. We are moving away from ambiguity to a more focused set of commercial drivers, prioritisation and thought-out delivery planning and timelines.

First, we start with something every business has: a list of problems and opportunities. This list needs to be framed from the perspective of customers, regulators, or internal sales or operations teams. Team members and not just leaders can propose their ideas, with bonuses for successful suggestions. This list will be long, but it's healthy to get as much down as possible. Next, we need to look at the expected returns from each problem statement. If we need to achieve regulatory compliance by the end of the year, that is still of value as it allows you to keep doing business. Some problems may be fascinating, but if they don't bring a material customer and revenue benefit, they shouldn't be prioritised. Ideally, we get to five or six force-ranked problems that make sense to solve.

Next, we start with the top three, and give them to teams who know this space well enough to develop potential solutions. This is known as *product discovery*. It is up to team members to figure out what needs to be done to satisfy customer needs and estimate costs and timelines, dropping the idea if the effort outweighs the results. Ideally, your teams will be thinking about a minimum viable product that would enhance revenue, hook customers, and iteratively improve to keep paying for itself. Teams should have their own domain expertise, get input from the right internal and external stakeholders, and do enough research (including prototyping) to be confident of their investment case.

User stories are succinct ways to describe better business outcomes, but they often aren't specific enough. You can't capture complex system behaviours and workflows on a post-it note! We get stuck between post-its which aren't detailed enough and lengthy requirements documents written in waffly management-speak. We need document discipline, insisting that the key people review it to ensure it captures all key points, and using objective readability measures like [Flesh-Kincaid](#) to make it succinct and easy to read. Good documents spell out explicitly what is in and out of scope, performance expectations, regulatory data, recovery scenarios and timelines. Clear, succinct and comprehensive documents are like gold dust - they help stakeholders to buy in and delivery teams to focus. The days of a few terse statements with some acceptance criteria are long gone for anyone serious about investment optimisation.

For example, my co-author, David Wilks was the Business Design Director for the current British system of electoral registration. It's working very successfully and we can all vote! But when he took over the project, it had a legacy of repeatedly missing milestones. How did he solve it? He simply got his team to redraft the design in decent, clear English. This halved the length of the document, presented it nicely, and suddenly everyone understood it and

signed it off! It was still comprehensive, with technical details and diagrams, but much shorter and more readable. This illustrates the power of a good requirements document!
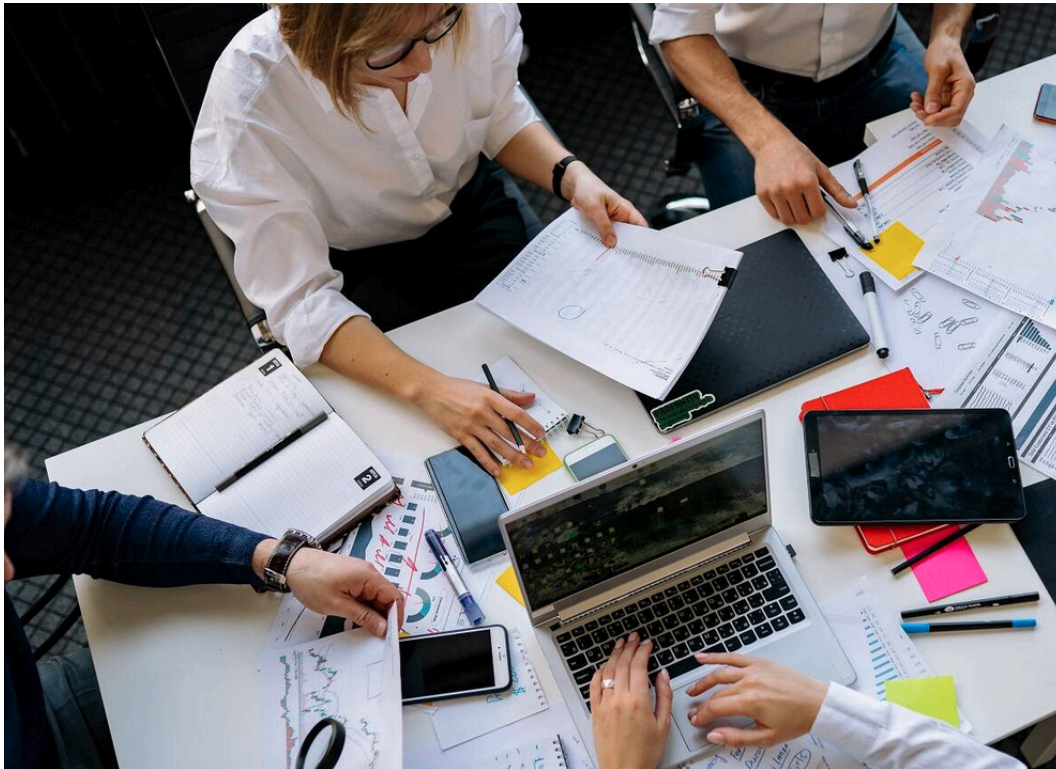
Fig 1: requirements documents often need cross-team collaboration

Finally, if the business case supports implementing the solution, the team needs a solid understanding of the user research and delivery challenges so that they can work to a delivery date that will get something of value to market. Leadership needs to support the team by making sure that they are single-threaded and not losing capacity by working on other deliverables. This is something that many firms, both big and small, fail to get right. Why would you dilute the capacity of people working on revenue growth to solve something that isn't as critical? Put other people together to work on business-as-usual and smaller enhancements or bug fixes so that delivery teams aren't diverted from the big ticket items. The authors have seen at first hand how smaller players have crushed bigger and better-funded rivals with sheer focus.

Only the most critical things to move the business forward should get delivery focus. Anything outside the top few items is simply deprioritised. Bright new ideas are added to the list of problems and ruthlessly assessed for commercial viability. Nothing should interfere with the work of a team building a new product, nor their ability to give frequent demos of their progress to relevant stakeholders: a valuable agile technique we want to keep.
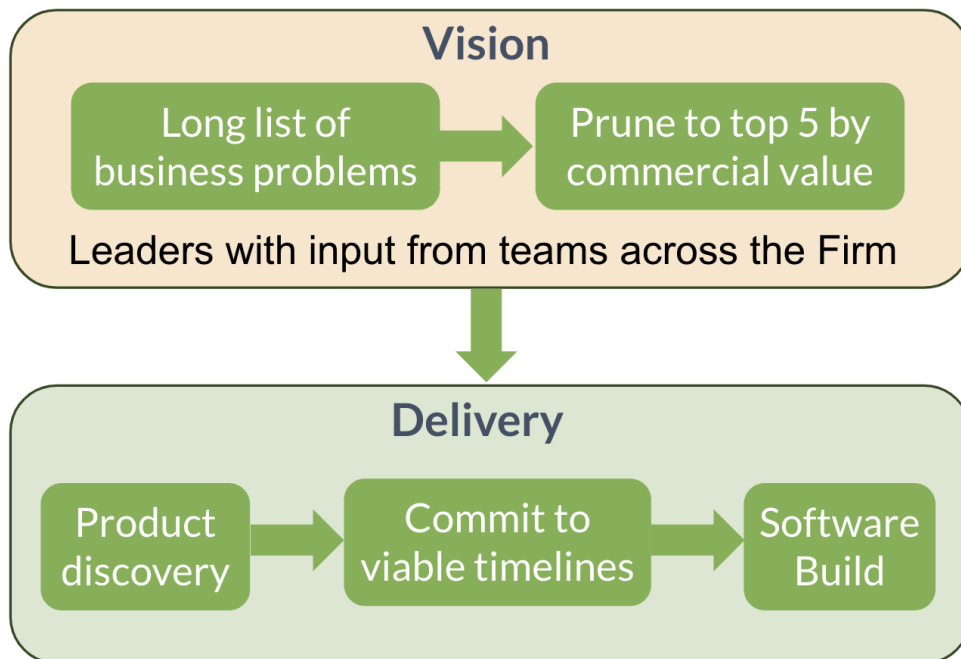
Fig 2: Postagile workflow

Conclusion

So, let's dispense with much of the unnecessary baggage of agile in favour of a more structured approach to financial value generation, driven by postagile principles. We start with a long list of business problems. We then prune down that list by working out which have the strongest commercial value. Giving the top handful of problems to teams to solve leads to stronger morale. Well-researched product discovery gives us confidence that customer needs will be met and viable delivery dates to make the team credible. You could sum up postagile as this: if we aren't going to earn from this new product, we won't spend!

References

Agile Manifesto Principles, agilemanifesto.org/principles.html.

Marty Cagan, *Transformed*, New Jersey: Wiley, 2024.

Jon Smart, et al., *Sooner safer happier: Antipatterns and patterns for business agility*, Portland, OR: Revolution, 2020.

Further information

Mark McKee is a founder of Agile Mind, based in London, UK. David Wilks is a big data and product delivery consultant. For more info, please contact mark.mckee@agilemind.uk