

A good pool player can win with a house cue.

[Michael Hay](#)



The title of this piece is a quote from a book on billiards my late father bought me. During high school we had a pool table and he wanted me to play often and to learn the basics, so he bought me a book. I suppose the thing I came away with is the quote's intent. Essentially, if you're a poor pool player an expensive cue doesn't make you a better player, just a poor player with an expensive cue. To become better you must practice and practice some more until you've honed your skills to their maximum potential. Once you're "good" you can still play well and win with the house cue. The thing I want you to take away is that once you've become the best the expensive cue isn't needed. Ironical!

Does this logic hold in Product Management? Yes, yes it holds, and the equivalent: You have to be great in Products before expensive tooling can

help, and once you're good you can use any tool. Yet it is fashionable today for Products teams to grasp for an expensive product roadmapping tool rather than become great at Products first. So like playing pool the poor Product Manager can construct a bad plan in an expensive tool. This makes the real requirement to practice and hone your skill until you're great. Once you're in the zone you'll always seem like you've done your homework because you have. You'll always be respected by your Engineering colleagues because you brought them along from discovery to product market fit. You'll have strong executive poise when talking about your product business because you invented your value proposition. Finally, just like pool once you're really great you could write stellar product plans in spreadsheets or even a plain text file.

It's not that I hate Products Management and Products Engineering tooling. In fact, I find tools like Jira and GitHub mandatory because they are essential to the Products workflow from initial commit to maximum differentiated product. Leaders in this space include [Atlassian](#), [GitHub](#) and [GitLab](#). These tools are deeply ingrained in how engineers prosecute their work. Strong source code and issue management ensures that as an engineer you're mindful of quality and expanding product capabilities to meet the demands of your users. Yet are these tools just for engineers?

GitHub: Reasonable for Product Managers!?

Over the past couple of years GitHub has progressed [Projects](#) as a part of their Project Management initiative (see image below).

As we looked at Projects we saw something fascinating. The structure of a Projects board is remarkably similar to feature boards in product and roadmapping tools. However unlike these roadmapping tools the cost is superior (\$0), it is adjacent to developer tooling, and to our delight there's room for improvement. With room for improvement we deeply believe in an opportunity to augment GitHub Projects with automation, GenAI, and structure solving real problems for Products teams. By adding these improvements to GitHub we think we're creating the metaphorical "house cue". In the spirit of pool, we intend to coach Products teams to be at their best for the infinite game of developing stellar products and services. When

they're done an expensive cue won't be needed.

Mediumroast for GitHub, release 1

Every journey starts with a first step and this is ours. In this first release we're intent to help our users on their product discovery journey. For us product (and service) discovery begins from a formulated hypothesis that is iteratively tested with target users and against competitors. We refer to these tests and their resulting artifacts as Interactions. From first hand experiences we recognize Interaction artifacts can be stranded, hidden from key constituents, disassociated from the Company they belong to, or even lost due to neglect.

As a result of this recognition, we're making the Product community aware of our first release combining an open-source product, [Mediumroast for GitHub](#), and our [professional services](#). What's available now is an open-source Command Line Interface (CLI) you can install to define Companies and store Interactions in a GitHub repository. There are also automations to enable basic reports on all Companies and on each Company.

There's more to come in our journey so stay tuned. And finally I want thank my founding team: [John Goodman](#), [Leonardo Del Riego](#), [Rizwan Muhammad](#) and [Raul Flores](#).