Make Site Go Brrr!

Or how to prioritise and articulate the need for tech improvements to non-technical stakeholders

Valeria Viana Gusmao - Head of Web, Strawberry



Do you prefer a supercar or a tractor?



Ferrari

Photo by Joshua Koblin on Unsplash



Beautiful Farm Country, Wayne County, Ohio July 2020.

Photo by Randy Fath on Unsplash

Compare the benefits?



Supercar

- Expensive
- Very fast
- Red
- Status symbol



Tractor

- Expensive
- Very slow
- Green
- Can farm fields

It depends on...

Needs

Do we need a supercar or a tractor now or in the future?

Constraints

Do we have enough money and storage for both? One? Can we only buy car parts?

Risks

What happens if we don’t get one or the other? Or neither of these?

Preference

Are there any trends or personal preferences for one or the other? Are tractors in style? Are Ferraris frowned upon?

Would you prefer to get $100 or $400?



$100

Photo by Maksym Tymchyk 🇺🇦 on Unsplash



$400

Photo by Ronak Ramnani on Unsplash

Easy! $400!

Same scale

Easy to compare since we’re comparing items on the same scale of the same units

Obvious benefits, same costs

The benefits are obvious, unless there are some hidden costs and risks

...but

Tech prioritisation is almost never a choice between $100 and $400

From roads to roadmap

Normalise the Scale

Convert different needs into a single language of impact

Maximise OKRs/KPIs

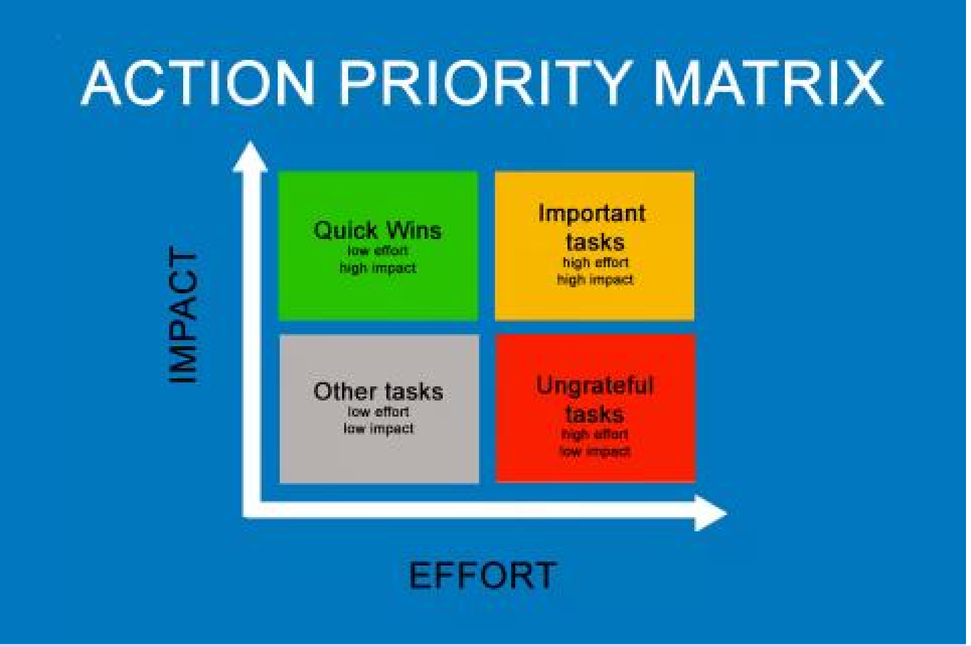
Filter every request through the lens of what actually moves the needle for the business

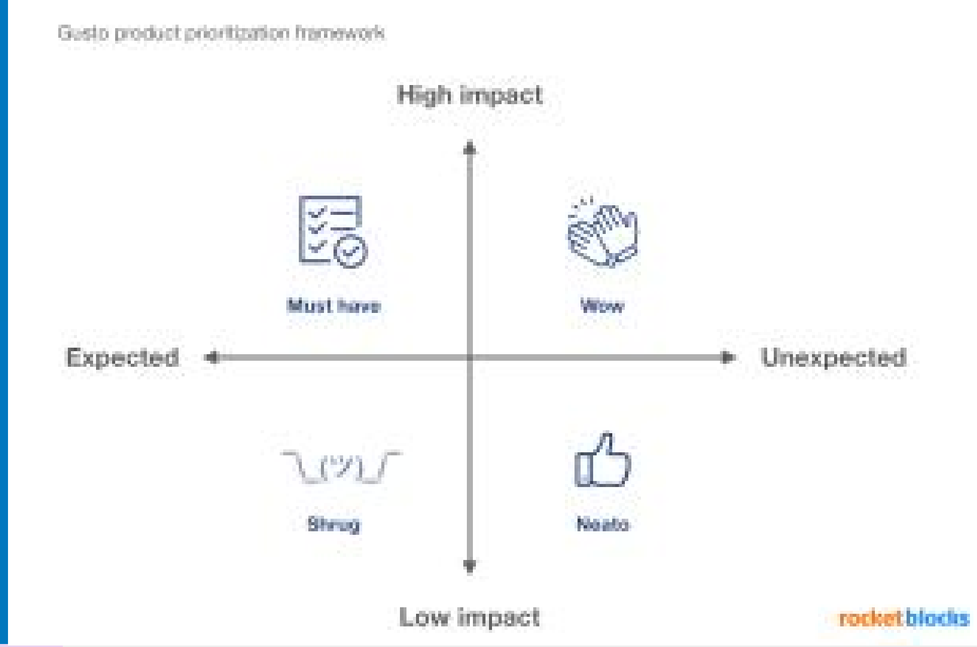
Balance the Price Tag

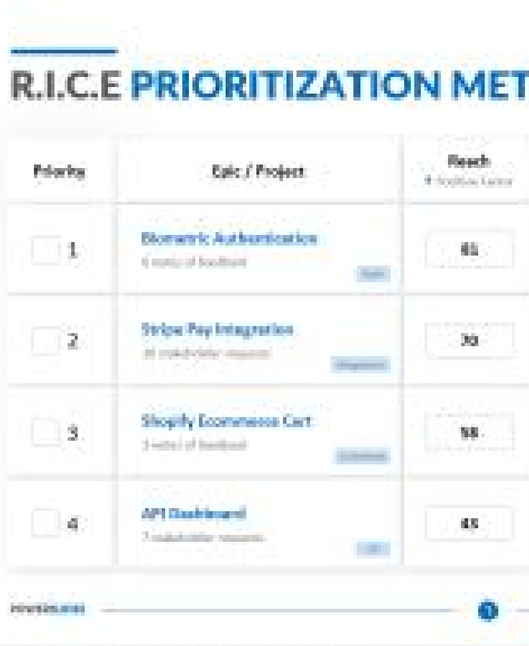
Weight those benefits against the reality of resources and time constraints

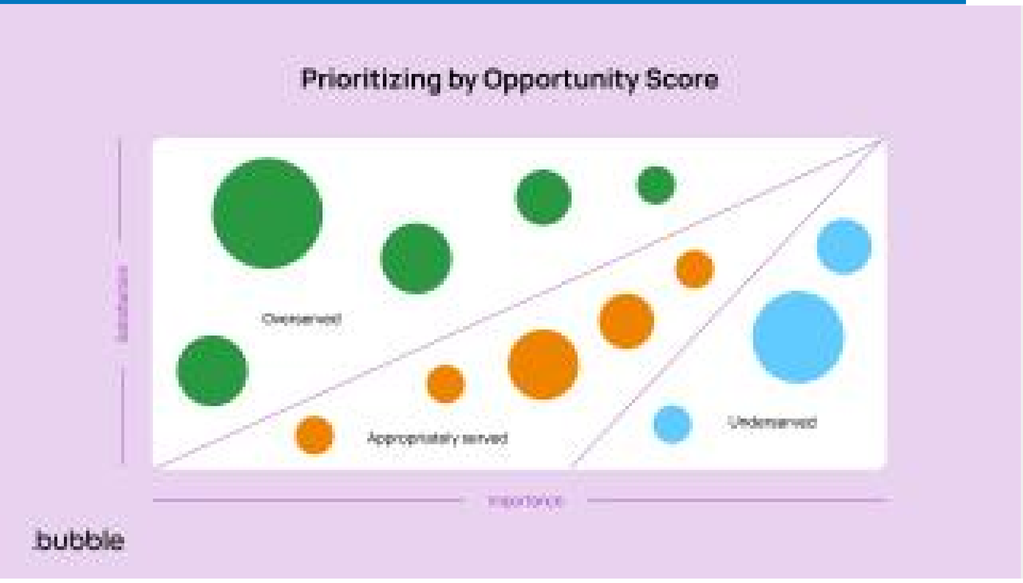
Evaluate risks

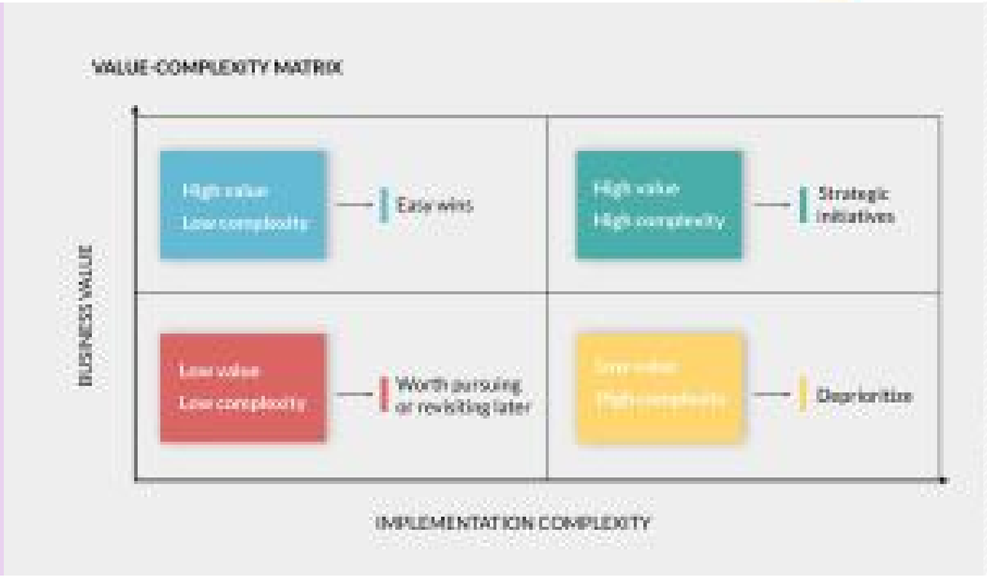
Explicitly account for what happens if we don’t act with a certain period of time - identifying the hidden costs of inaction













Product managers have many different prioritisation techniques up their sleeves...

How can we help prioritise?

Estimate Effort

As tech specialists we would be doing the work and are the ones who would be able to make the closest **estimate** to how much the implementation would actually take

Highlight Risks

Some risks that are obvious to you might not be visible to others at all, so make sure to highlight the risks you see and how they can be mitigated.

Common risks are **reliability**, **compliance** and **increased costs**

Articulate impact

To be able to prioritise when to do what stakeholders need to understand the **impact** and **urgency.**

What KPIs would this affect, how and when? Does it compound over time?

The Phoenix Project: A Novel About IT, DevOps, and Helping Your Business Win

Novel by Gene Kim, George Spafford, and Kevin Behr



Make site go brrr!

Example

We want to implement that new cool framework we discovered, it would make our hydration speed blazing fast and it’s going to be so cool!

5% more orders

Example

We estimated that ~5% of our users churn due to low website performance.

We have a PoC that shows significant improvement by using a new frontend framework. It would take 3 developers **2 weeks** to take this to production.

We do not see any additional risks as our code is well covered with tests and we are confident that we can catch all critical regressions during development.

If we do it before the next month’s campaign we will be able to maximise the profits.

Want to train more?

Reflect

Keep trying, iterating and reflecting on the decisions. Was it the right call? Did we see it coming? What would you do differently now?

Ask

Talk to the stakeholders. Ask them how they prioritise, what is important to them and why.

Monitor

Set SLIs and monitor your technical platform. It’s easy to come up with improvements based on the metrics you’re trying to improve than the other way around.

Train

Ask AI to quiz you on hypothetical cases. I found Gemini and Claude to be quite good at it. Or ask them to role-play a planning session where you present an improvement to help prepare for the real one.



Valeria Viana Gusmao

Creator of Hounty | Head of Web

@Strawberry | Building systems for people

Let’s connect!

Thank you