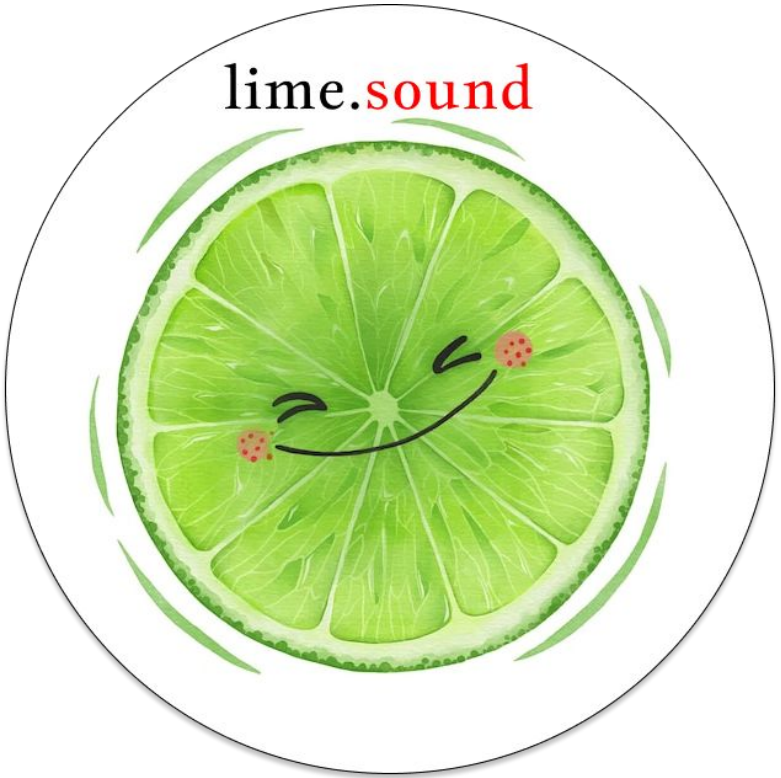
Requirements prioritisation

IMT2019525 VIJAY JAISANKAR

Lime



Lime: An end-to-end music creation, hosting, and promotion platform for budding creators.

Mission statement: Harness the power of communities and make your sounds heard.

Vision statement: To be the top preferred music platform for any new creator, anywhere.

Requirements prioritisation

Source

https://www.businessanalystlearnings.com/blog/2016/8/18/a-list-of-requirements-prioritization-techniques-you-should-know-about

Ranking

In this method, we give each requirement a different numerical value based on its importance.

- This method can be useful in limited contexts after Lime has matured in the market
- Users can vote on the new feature they like the most, and that can be taken up by the development team
- **However, as this method only works for one stakeholder and as the product is early-stage, we won't be using this method.**

Grouping

In this method, requirements are grouped into critical, moderate, and optional priorities.

- This method seems more promising than ranking, as the discretisation leads to less confusing classification between requirements
- However, creating the mental map or a scoring metric for the priorities will be a hassle and communicating the same adds to the complexity.
- **Hence, although this method is better than Ranking, we can do better for Lime.**

MoScoW

Instead of numbers, here, requirements are grouped as follows:

- MUST (Mandatory)
- SHOULD (Of high priority)
- COULD (Preferred but not necessary)
- WOULD (Can be postponed and suggested for future execution)

Why MoScoW is the best technique for Lime

- This is the most intuitive method for the most important stakeholders to vote on - as there is no numerical thresholding to rely on, and the notions of MoScoW are natural to all people in the business.
- Getting the inputs for the same are also easy - through community meetings and forms for the artists and users alike.
- Some parts of the MoScoW are directly transferable from the licenses and SLAs.
- All of the outputs of this technique will directly contribute to timely and organised building of Lime and can be implementable in an Agile development lifecycle.
- **Hence, of the requirement prioritisation methods we've seen, MoScoW is the most optimal, and hence, we will use this method for Lime.**

Methodology

How Lime implements the MoScoW technique

Requirements Gathering

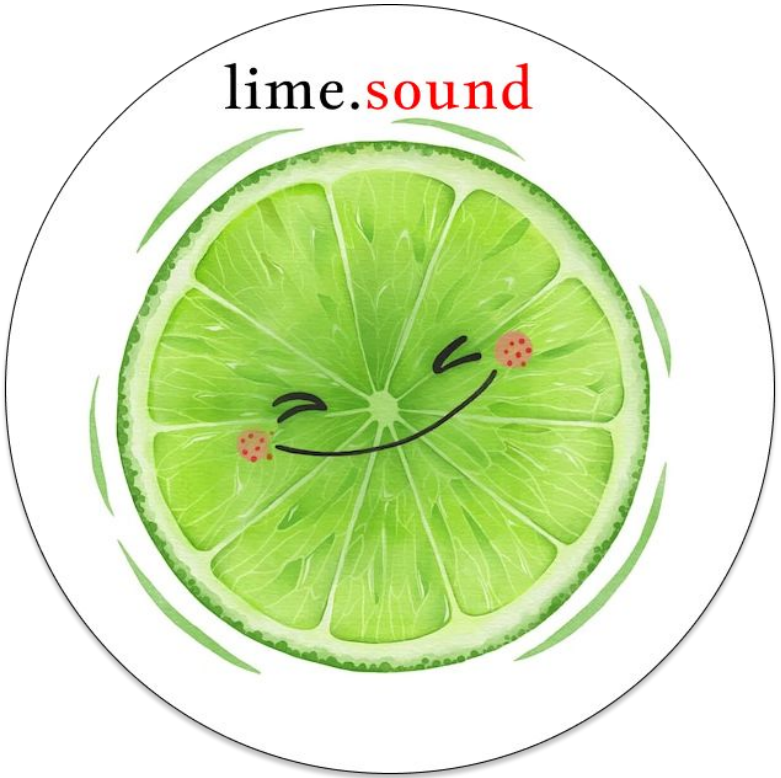
From different stakeholders

- The MUST requirements directly follow from the Lean canvas and value canvas, and the market research
- We can then gather SHOULD and COULD requirements directly from the site and on Github discussions
- We will also hold frequent community meet-ups, which can serve as a faucet for COULD and WOULD requirements, social media, online communities, and Github issues will also contribute to these.

Using the requirements effectively

Incremental development of the product

- We can use Github releases and create separate labels for each schema of the MoScoW framework.
- We can also track these requirements and create sprint plans using Github project boards
- We can then manage mappings between releases and requirements in a transparent manner and handle scope changes effectively, by keeping discussions and meetings.



Thank you!

IMT2019525 Vijay Jaisankar