# Prioritizing and formulating research questions

In an engineering project, often research questions arise in order to determine a development route towards a product that meets the requirements that have been listed in conjunction with the project stakeholders. Setting up your research requires to first prioritize the questions you have, and we will use the MoSCoW method to achieve that. Then you will formulate your research questions such that your research activity reveals useful and reliable information, by applying the SMART criteria.

## The MoSCoW method

*This text is directly adapted from <https://en.wikipedia.org/wiki/MoSCoW_method>, which is available under the [Creative Commons Attribution-ShareAlike License](https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License).*

During the course of a project it is usually not feasible to cover all of the requirements desired by the client. For that reason, engineers deploy some kind of prioritization method. We will use the MoSCoW method, the same technique can also be applied to prioritizing research questions.

The term MoSCoW itself is an acronym derived from the first letter of each of four prioritization categories (*Must have*, *Should have*, *Could have*, and *Won't have*), with the interstitial *O*s added to make the word pronounceable.

All requirements are important, but they are prioritized to deliver the greatest and most immediate business benefits early. Developers will initially try to deliver all the Must have, Should have and Could have requirements but the Should and Could requirements will be the first to be removed if the delivery timescale looks threatened. The categories are typically understood as:

**Must have** - Requirements labeled as Must have are critical to the project in order for it to be a success. Regarding research questions: Why are you doing this research? What benefit will your findings have to the project? If even one Must have is not included, the project delivery should be considered a failure (note: requirements can be downgraded from Must have, by agreement with all relevant stakeholders; for example, when new requirements are deemed more important). MUST can also be considered an acronym for the Minimum Usable SubseT.

**Should have** - Requirements labeled as Should have are important but not necessary. While Should have requirements can be as important as Must have, they are often not as time-critical or there may be another way to satisfy the requirement, so that it can be held back until a future delivery timebox.

**Could have** - Requirements labeled as Could have are desirable but not necessary, and could improve user experience or customer satisfaction for little development cost. These will typically be included if time and resources permit.

**Won't have** (this time) - Requirements labeled as Won't have been agreed by stakeholders as the least-critical or not appropriate at that time. As a result, Won't have requirements are not planned into the project schedule. Won't have requirements are either dropped or reconsidered for inclusion in a later timebox. (Note: occasionally the term Would like to have is used; however, that usage is incorrect, as this last priority is clearly stating something is outside the scope of delivery).

## Robot car research question prioritization example

|  |  |  |
| --- | --- | --- |
| Q1 | What power source should be applied? | S |
| Q2 | How can the line be detected? | M |
| Q3 | How can the robot car speed be increased? | C |
| Q4 | What controller should be applied? | W |

## SMART criteria

*This text is loosely based on <https://en.wikipedia.org/wiki/SMART_criteria>, which is available under the [Creative Commons Attribution-ShareAlike License](https://en.wikipedia.org/wiki/Wikipedia:Text_of_Creative_Commons_Attribution-ShareAlike_3.0_Unported_License).*

SMART is a mnemonic/acronym, giving criteria to guide in the setting of goals and objectives. The principal advantage of SMART objectives is that they are easier to understand and to know when they have been done. A requirement is nothing more than an expected outcome of the project, and it can be made SMART. Also a research questions need to become SMART in order to reveal reliable information. The letters S and M generally mean **specific** and **measurable**. Possibly the most common version has the remaining letters referring to **achievable**, **relevant** and **time-bound**.

Ideally speaking, each requirement should be:

**Specific** – Requirements should be specific and not generic. They should not be open to mis-interpretation when read by others. This is the most important attribute to get correct. Your research question should be specific as well, the most common mistake is to set your topic too broadly.

**Measurable** – Requirements should be quantifiable in order to determine its completion. You should avoid signing up for any requirement that cannot be verified as complete. These are especially risky when you use non-quantitative terms (best, optimal, fastest) for acceptance criteria. Also make sure that your research question is measurable. You must be able to support your argument logically with experiments, studies, and other evidence that aren’t just anecdotal.

**Achievable** – Ensure that the requirement is realistically able to be achieved given existing circumstances and available resources. Also your research question needs to be realistic.

**Relevant** – Requirements and research questions should be relevant to a project, check with MoSCoW.

**Time-bound** – Each requirement or research question should be time-bound or specify by when or how fast a requirement needs to be completed or executed.

## Robot car research question prioritization example

|  |  |  |
| --- | --- | --- |
| Q1 | What battery allows to run the robot car at a speed of 10cm/s for 1 hour on a full charge and can be bought on the project budget?  AND define criteria to select the battery? | S |
| Q2 | What line sensor(s) enables the robot car to follow a black line of 1 cm width on a white surface with an accuracy of .5cm at a speed of 10cm/s, can be interfaced to the MCU and can be bought on the project budget?  AND define criteria to select the sensor? | M |
| Q3 | How should the robot car be adapted to follow a black line of 1 cm width on a white surface with an accuracy of .5cm at 20cm/s for 1 hour on a full charge?  subQ3.1 What hardware changes enable the robot car to achieve 20cm/s? subQ3.1.1 What motors enable the robot car to achieve 20cm/s?  subQ3.1.1 What battery allows the car to run at 20cm/s for 1 hour on a full charge?  subQ3.2 What software changes are needed to follow a black line of 1 cm width on a white surface with an accuracy of .5cm at 20cm/s?  subQ3.2.1 Is a controller change required to accommodate the SW change?  subQ3.3 What battery change …. | C |
|  |  |  |

## Examples of weak questions

How can the robot car run on a battery?

How can the robot car run as fast as possible?

How can the robot car follow a line?