

HANDBOOK DSDM

LINK TO SITE: <https://www.agilebusiness.org/content/moscow-prioritisation>

MoSCoW Prioritisation

10.1 Introduction

In a DSDM project where time has been fixed, it is vital to understand the relative importance of the work to be done in order to make progress and keep to deadlines. Prioritisation can be applied to requirements/User Stories, tasks, products, use cases, acceptance criteria and tests, although it is most commonly applied to requirements/ User Stories.

MoSCoW is a prioritisation technique for helping to understand and manage priorities. The letters stand for:

- **M**ust Have
- **S**hould Have
- **C**ould Have
- **W**on't Have this time

The use of MoSCoW works particularly well on projects. It also overcomes the problems associated with simpler prioritisation approaches which are based on relative priorities:

- The use of a simple high, medium or low classification is weaker because definitions of these priorities are missing or need to be defined. Nor does this categorization provide the business with a clear promise of what to expect. A categorisation with a single middle option, such as medium, also allows for indecision
- The use of a simple sequential 1,2,3,4... priority is weaker because it deals less effectively with items of similar importance. There may be prolonged and heated discussions over whether an item should be one place higher or lower

The specific use of Must Have, Should Have, Could Have or Won't Have this time provides a clear indication of that item and the expectations for its completion.

10.2 The MoSCoW Rules

10.2.1 Must Have

These provide the Minimum Usable SubseT (MUST) of requirements which the project guarantees to deliver. These may be defined using some of the following:

- No point in delivering on target date without this; if it were not delivered, there would be no point deploying the solution on the intended date
- Not legal without it
- Unsafe without it
- Cannot deliver a viable solution without it

Ask the question 'what happens if this requirement is not met?' If the answer is 'cancel the project – there is no point in implementing a solution that does not meet this requirement', then it is a Must Have requirement. If there is some way around it, even if it is a manual and painful workaround, then it is a Should Have or a Could Have requirement. Categorising a requirement as a Should Have or Could Have does not mean it won't be delivered; simply that delivery is not guaranteed.

10.2.2 Should Have

Should Have requirements are defined as:

- Important but not vital
- May be painful to leave out, but the solution is still viable
- May need some kind of workaround, e.g. management of expectations, some inefficiency, an existing solution, paperwork etc. The workaround may be just a temporary one

One way of differentiating a Should Have requirement from a Could Have is by reviewing the degree of pain caused by the requirement not being met, measured in terms of business value or numbers of people affected.

10.2.3 Could Have

Could Have requirements are defined as:

- Wanted or desirable but less important
- Less impact if left out (compared with a Should Have)

These are the requirements that provide the main pool of contingency, since they would only be delivered in their entirety in a best case scenario. When a problem occurs and the deadline is at risk, one or more of the Could have's provide the first choice of what is to be dropped from this timeframe.

10.2.4 Won't Have this time

These are requirements which the project team has agreed will not be delivered (as part of this timeframe). They are recorded in the Prioritised Requirements List where they help clarify the scope of the project. This avoids them being informally reintroduced at a later date. This also helps to manage expectations that some requirements will simply not make it into the Deployed Solution, at least not this time

around. Won't Haves can be very powerful in keeping the focus at this point in time on the more important Could Haves, Should Haves and particularly the Must Haves.

10.3 MoSCoW Relating to a Specific Timeframe

In a traditional project, all requirements are treated as Must Have, since the expectation is set from the start that everything will be delivered and that typically time (the end date) will slip if problems are encountered. DSDM projects have a very different approach; fixing time, cost and quality and negotiating features. By the end of Foundations, the end dates for the project and for the first Project Increment are confirmed.

In order to meet this commitment to the deadline, DSDM projects need to create contingency within the prioritised requirements. Therefore the primary focus initially is to create MoSCoW priorities for the project. However, when deciding what to deliver as part of the Project Increment, the next focus will be to agree MoSCoW priorities for that Increment. So at this point, a requirement may have two priorities; MoSCoW for the project and MoSCoW for the Increment. Finally, when planning a specific Timebox (at the start of each Timebox) the Solution Development Team will allocate a specific priority for the requirements for this Timebox. At this point, the majority of requirements are Won't Have (for this Timebox). Only requirements that the Solution Development Team plan to work on in the development timebox are allocated a Must Have, Should Have or Could Have priority.

Therefore requirements may have three levels of priority:

- MoSCoW for the project
- MoSCoW for the Project Increment
- MoSCoW for this Timebox

For example:

even if a Must Have requirement for an IT solution is the facility to archive old data, it is very likely that the solution could be used effectively for a few months without this facility being in place. In this case, it is sensible to make the archive facility a Should Have or a Could Have for the first Project Increment even though delivery of this facility is a Must Have before the end of the project. Similarly, a Must Have requirement for a Project Increment may be included as a Should Have or a Could Have (or a Won't Have) for an early Timebox.

It is important that the bigger picture objectives (completion of the Project Increment and delivery of the project) are not forgotten when working at the Timebox level.

One simple way to deal with this is to create a separate Timebox PRL, a subset of the project PRL that is specifically associated with an individual Timebox and leave the priorities unchanged on the main PRL for the project.

10.4 Ensuring effective prioritisation

10.4.1 Balancing the priorities

When deciding the effort allocated for Must Have requirements, remember that anything other than a Must Have is, to some degree, contingency, since the Must Haves define the Minimum Usable SubseT which is guaranteed to be delivered.

DSDM recommends:

- Getting the percentage of project/Project Increment Must Haves (in terms of effort to deliver) to a level where the team's confidence to deliver them is high – typically no more than 60% Must Have effort
 - Agreeing a pool of Could Haves for the project/Project Increment that reflects a sensible level of contingency - typically around 20% Could Have effort.
- Creating a sensible pool of Could Haves sets the correct expectations for the business from the start – that these requirements/User Stories may be delivered in their entirety in a best case scenario, but the primary project/Project Increment focus will always be on protecting the Must Haves and Should Haves

This spread of priorities provides enough contingency to ensure confidence in a successful project outcome.

NB When calculating effort for a timeframe, Won't Haves (for this timeframe) are excluded.

DSDM's recommendations reflect a typical project scenario. The important thing to make MoSCoW work is to have some visible flexibility in the level of requirements which must be delivered.

The safe percentage of Must Have requirements, in order to be confident of project success, is not to exceed 60% Must Have effort.

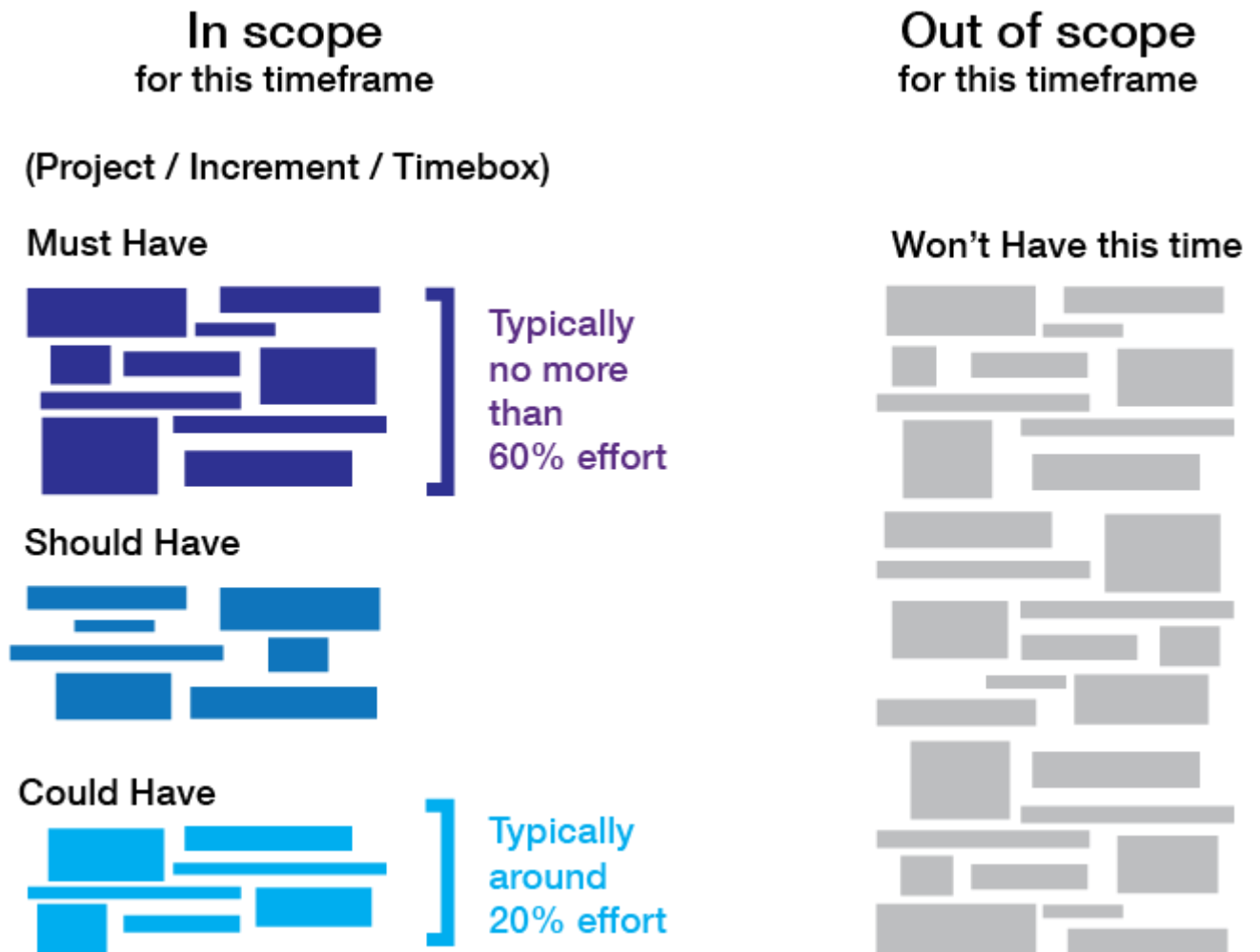


Figure 10a: MoSCoW – balancing priorities

Levels of Must Have effort above 60% introduce a risk of failure, unless the team are working in a project where all of these criteria are true:

- Estimates are known to be accurate
- The approach is very well understood
- The team are “performing” (based on the Tuckman model)
- The environment is understood and low-risk in terms of the potential for external factors to introduce delays

In some circumstances the percentage of Must Have effort may be significantly less than 60%. However this can be used to the benefit of the business, by providing the greatest possible flexibility to optimise value delivered across a larger proportion of Should Haves.

The exact split of effort between Musts, Shoulds, and Coulds is down to each project team to agree, although DSDM also recommends creating a sensible pool of Could

Haves, typically around 20% of the total effort. Effective MoSCoW prioritisation is all about balancing risk and predictability for each project.

10.4.2 Agreeing up front how priorities will work

DSDM defines what the different priorities mean – the MoSCoW Rules. But whereas the definition of a Must Have is not negotiable, the difference between a Should Have and a Could Have can be quite subjective. It is very helpful if the team agree, at the start of their project, how these lower level priorities will be applied. Understanding in advance some objective criteria that separate a Should Have from a Could Have and ensuring that all roles on the project buy into what has been agreed can avoid much heated discussion later. Look for defined boundaries that decide whether a requirement is a Should Have or a Could Have?

For example:

At what point does the number of people impacted raise a Could Have to a Should Have?
Or, What

value of benefits would justify dropping this requirement from a Should Have to a Could Have?

Ideally this agreement is reached before the requirements are captured.

10.4.3 When to prioritise

Every item of work has a priority. Priorities are set before work commences and the majority of this prioritisation activity happens during Foundations. However, priorities should be kept under continual review as work is completed. As new work arises, either through introduction of a new requirement or through the exposure of unexpected work associated with existing requirements, the decision must be made as to how critical it is to the success of the current work using the MoSCoW rules. When introducing new requirements, care needs to be taken not to increase the percentage of Must Have requirement effort beyond the agreed project level. The priorities of uncompleted requirements should be reviewed throughout the project to ensure that they are still valid. As a minimum, they should be reviewed at the end of each Timebox and each Project Increment.

10.4.4 Discussing and reviewing priorities

Any requirement defined as a Must Have will, by definition, have a critical impact on the success of the project. The Project Manager, Business Analyst and any other member of the Solution Development Team should openly discuss requirements prioritised as Must Have where they are not obvious Must Haves (“Without this would we cancel the project/increment?”); it is up to the Business Visionary or their empowered Business Ambassador to explain why a requirement is a Must Have.

The escalation of decision-making processes should be agreed early on, e.g. Business Ambassador and Business Analyst to Business Visionary to Business

Sponsor, and the level of empowerment agreed around decision-making at each level.

At the end of a Project Increment, all requirements that have not been met are re-prioritised in the light of the needs of the next Increment. This means that, for instance, a Could Have that is not met in one Increment may be reclassified subsequently as a Won't Have for the next Increment, because it does not contribute enough towards the business needs to justify its inclusion. However, it could just as easily become a Must Have for the next Increment, if its low priority in the first Increment was based on the fact it was simply not needed in the first Solution Increment.

10.5 Using MoSCoW to Manage Business Expectations

The MoSCoW rules have been defined in a way that allows the delivery of the Minimum Usable SubseT of requirements to be guaranteed. Both the Solution Development Team and those to whom they are delivering share this confidence because the high percentage effort of Shoulds and Coulds provides optimum contingency to ensure delivery of the Must Haves.

The business roles can certainly expect more than delivery of only the Must Haves. The Must Haves are guaranteed but it is perfectly reasonable for the business to expect delivery of more than the Minimum Usable SubseT in the timeframe, except under the most challenging of circumstances.

DSDM's recommendation to create a sensible pool of Could Have contingency – typically around 20% of the total project/increment effort - identifies requirements that are less important or which have less impact if not delivered, in order to protect the more important requirements. This approach implies that the business can reasonably expect the Should Have requirements to be met, in addition to all of the Must Haves. It also implies that in a best case scenario, the Could Have requirements would also be delivered.

The Solution Development Team cannot have the confidence to guarantee delivery of all the Must Have, Should Have and Could Have requirements, even though these have all been estimated and are included in the plan. This is because the plan is based on early estimates and on requirements which have not yet been analysed in low-level detail. Applying pressure to a team to guarantee delivery of Musts, Shoulds and Coulds is counter-productive. It usually results in padded estimates which give a false perception of success. "We always achieve 100% (because we added significant contingency to our figures)".

So, combining sensible prioritisation with timeboxing leads to predictability of delivery and therefore greater confidence. This also protects the quality of the solution being delivered. Keeping project metrics to show the percentage of Should Haves and Could Haves delivered on each Project Increment or Timebox will either re-enforce

this confidence, if things are going well, or provide an early warning of problems, highlighting that some important (but not critical) requirements may not be met at the project level.

10.6 How does MoSCoW Relate to the Business Vision

10.6.1 The Business Sponsor's perspective

The starting point for all projects is the business vision. Associated with the business vision are a set of prioritised requirements that contribute to delivery of the vision. Also associated with the business vision is a Business Case that describes the project in terms of what value it will deliver back to the business. Depending on the organization, this Business Case may be an informal understanding or it may be defined formally, showing what Return On Investment (ROI) is expected in order to justify the cost of the project.

The MoSCoW priorities are necessary to understand the Minimum Usable SubseT and the importance of individual requirements. The Business Visionary must ensure that the requirements are prioritised, evaluated in business terms, and delivered to provide the ROI required by the Business Case, in line with the business vision.

10.7 Making MoSCoW Work

Requirements are identified at various levels of detail, from a high-level strategic viewpoint (typically during Feasibility) through to a more detailed, implementable level (typically during Evolutionary Development). Highlevel requirements can usually be decomposed to yield a mix of sub-requirements, which can then be prioritised individually. This ensures the flexibility is maintained, so that if necessary, some of the detailed less important functionality can be dropped from the delivered solution to protect the project deadline.

It is this decomposition that can help resolve one of the problems that often confront teams: that all requirements appear to be Must Haves.

If all requirements were genuinely Must Haves, then the flexibility derived from the MoSCoW prioritisation would no longer work. There would be no lower priority requirements to be dropped from the deliverables to keep a project on time and budget. This goes against the DSDM ethos of fixing time and cost and flexing features (the triangles diagram in the Philosophy and Fundamentals chapter). Believing everything is a Must Have is often symptomatic of insufficient decomposition of requirements.

Remember that team members may cause scope creep by working on "interesting" things rather than the important things. MoSCoW can help avoid this.

10.8 Tips for Assigning Priorities

1. Ensure that the business roles, in particular the Business Visionary and the Business Analyst, are fully up to speed as to why and how DSDM prioritises

requirements.

2. Consider starting with all requirements as Won't Haves, and then justify why they need to be given a higher priority.
3. For each requirement that is proposed as a Must Have, ask: 'what happens if this requirement is not met?' If the answer is 'cancel the project; there is no point in implementing a solution that does not meet this requirement', then it really is a Must Have. If not, then decide whether it is Should Have or a Could Have (or even a Won't Have this time)
4. Ask: 'if I come to you the night before Deployment and tell you there is a problem with a Must Have requirement and that we can't deliver it – will you stop the Deployment?' If the answer is 'yes' then this is a Must Have requirement. If not, decide whether it is Should Have or a Could Have.
5. Is there a workaround, even if it is a manual one? If a workaround exists, then it is not a Must Have requirement. When determining whether this is a Should Have or a Could Have requirement, compare the cost of the workaround with the cost of delivering the requirement, including the cost of any associated delays and any additional cost to implement it later, rather than now.
6. Ask why the requirement is needed – for this project and this Project Increment.
7. Is this requirement dependent on any others being fulfilled? A Must Have cannot depend on the delivery of anything other than a Must Have because of the risk of a Should Have or Could Have not being delivered.
8. Allow different priorities for acceptance criteria of a requirement.

For example:

'The current back-up procedures need to ensure that the service can be restored as quickly as possible.' How quick is that? Given enough time and money, that could be within seconds. A smarter definition

would be to say it Should happen within four hours, but it Must happen within 24 hours.

9. Can this requirement be decomposed? Is it necessary to deliver each of these elements to fulfil the requirement? Are the decomposed elements of the same priority as each other?
10. Tie the requirement to a project objective. If the objective is not a Must Have, then probably neither is the requirement relating to it.
11. Does the priority change with time? For example, for an initial release a requirement is a Should Have, but it will become a Must Have for a later release.
12. Prioritise testing, using MoSCoW.
13. Use MoSCoW to prioritise your To Do list. It can be used for activities as well as requirements.

10.9 Summary

MoSCoW (Must Have, Should Have, Could Have, Won't Have this time) is primarily used to prioritise requirements, although the practice is also useful in many other areas.

On a typical project, DSDM recommends no more than 60% effort for Must Have requirements on a project, and a sensible pool of Could Haves, usually around 20%

effort. Anything higher than 60% Must Have effort poses a risk to the success and predictability of the project, unless the environment and any technology is well understood, the team is well established and the external risks minimal.