How to Prioritize Work Using MoSCoW and RICE



Prioritise against an objective

A method helps structure a decision, while the project still needs evidence and authority.

Define the purpose

Clarify the outcome, planning period, constraints and decision owner before assigning categories or scores. Priorities are meaningful only in that context.

Service-app case

A field-service app aims to reduce avoidable booking calls. Several changes compete for a limited release capacity, including validation, reminders and cancellation.

Use the lab

MoSCoW helps discuss necessity for a stated period. RICE compares estimated opportunity under consistent assumptions. Use each for the question it can support.

TAKE IT TO YOUR PROJECT

Write the outcome and time horizon for your prioritisation.

Prepare comparable options

An uneven list makes a tidy ranking misleading.

Define the work

Describe each option's user result, scope, dependencies, evidence and approximate size. Split or clarify items that are too broad to compare sensibly.

App case

"Improve the service" is too large beside "add cancellation confirmation." Define the actual useful slice and the work needed for it.

Check the boundary

Mandatory commitments, defects, enabling work and discretionary features may need different treatment. Record why an option is being considered now.

TAKE IT TO YOUR PROJECT

Rewrite one vague candidate as a defined result.

Set the MoSCoW timebox

A priority applies to a specified commitment or planning period.

Name the horizon

Use Must Have, Should Have, Could Have and Won't Have this time within an explicit timebox or release context. Define the decision criteria with the relevant people.

App case

The pilot needs a valid booking path in the first release. Other improvements are considered against that release's purpose, workarounds and capacity.

Keep the meaning

Won't Have this time records a current boundary, not necessarily a permanent rejection. Revisit it through the agreed future decision process.

TAKE IT TO YOUR PROJECT

State the release or timebox to which your categories apply.

Identify genuine Must Haves

A Must needs a defensible reason under the agreed objective.

Test necessity

Ask whether the intended solution is viable without the requirement, whether an acceptable workaround exists and which binding condition makes it essential.

App case

A customer must select a valid available slot for the booking service to work. A preferred screen colour is unlikely to have the same necessity.

Protect flexibility

If every item is Must, the categorisation offers little room to respond when estimates or conditions change. Investigate the underlying needs and minimum usable scope.

TAKE IT TO YOUR PROJECT

Challenge one Must by asking what happens if it is absent.

Separate Should, Could and Won't

Useful distinctions make the scope trade-off discussable.

Should and Could

A Should is important but the release may remain viable with an acceptable workaround. A Could is desirable and provides flexibility if capacity is constrained.

App case

Self-service cancellation may be Should when staff can process it temporarily. A cosmetic preference may be Could. A new event-sales module is Won't for this booking release.

Agree the criteria

Use value, consequence, workarounds and timing rather than the requester's influence alone. Make the current boundary visible.

TAKE IT TO YOUR PROJECT

Record the workaround supporting one Should classification.

The All-in-One PM Templates Bundle

940+ Premium Templates • One Payment • Lifetime Access

$49

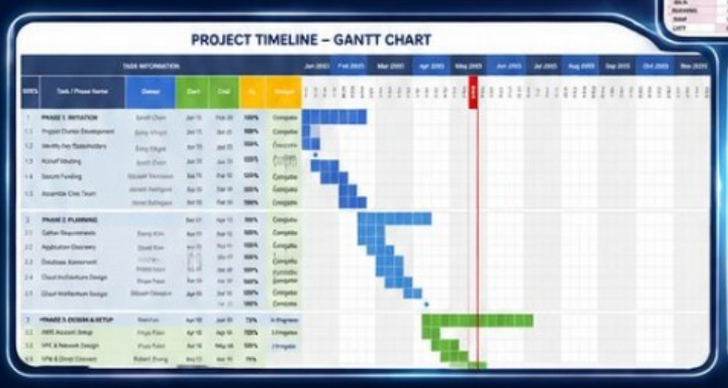
SAVE 94%

Worth $3,300+











EXCEL & GOOGLE SHEETS | PROFESSIONAL & RELIABLE | AUTOMATED DASHBOARDS | SAVE TIME & STAY ON TRACK | INSTANT ACCESS DOWNLOAD

GET LIFETIME ACCESS

Check MoSCoW against capacity

Categories do not make the required work fit automatically.

Estimate the commitment

Review effort, dependencies, uncertainty and actual capacity for the chosen period. Preserve enough flexibility for learning and variation under the approach being used.

App case

The Must set already consumes more capacity than available. The team needs a smaller viable scope, a changed date, more credible capability or another authorised response.

Use the conversation

Discuss the consequence of each choice with the relevant owner. Relabelling work cannot solve an infeasible commitment.

TAKE IT TO YOUR PROJECT

Compare your Must workload with real delivery capacity.

Understand the RICE factors

Use one consistent basis before multiplying estimates.

Reach and impact

Reach estimates how many relevant users or events are affected in the stated period. Impact estimates the effect per unit using an agreed scale.

Confidence and effort

Confidence reflects strength of the supporting estimates, expressed consistently as a fraction in the formula. Effort estimates the required work, using one unit across options.

App case

This guide uses users reached per quarter and person-months of effort. It treats all four factors as estimates with documented assumptions.

TAKE IT TO YOUR PROJECT

Write the unit and evidence source for each factor.

Calculate the RICE score

The arithmetic is simple; the estimates require judgement.

Use the formula

RICE = Reach x Impact x Confidence / Effort, with effort greater than zero. In this example, 80% confidence is entered as 0.80.

Define the scale

Use the same impact scale across candidates. Intercom's familiar scale includes 3, 2, 1, 0.5 and 0.25 for successively smaller estimated effects.

Check the output

The score is a comparative aid under the chosen units and assumptions. It is not a measured financial return or a guarantee of user value.

TAKE IT TO YOUR PROJECT

Calculate a candidate with its confidence entered as a fraction.

Work candidate A

Make each factor understandable before accepting its score.

State the estimates

A booking-validation improvement is estimated to reach 100 users this quarter, impact 1, confidence 80% and effort 2 person-months.

Calculate

100 x 1 x 0.80 / 2 = 40. Keep the evidence and uncertainty beside the score so reviewers can challenge its basis.

Read the result

Another team using different units or scales would produce a different number. Compare candidates within the same agreed model.

TAKE IT TO YOUR PROJECT

Explain the largest source of uncertainty in candidate A.

Compare three candidates

Use a common period, unit and impact scale for the set.

The options

A validation: reach 100, impact 1, confidence 0.80, effort 2. B reminders: 300, 0.5, 0.60, 3. C cancellation: 80, 2, 0.90, 2.

The scores

A scores 40. B scores 30. C scores 72. The numerical order is C, A, B under these particular assumptions.

Use the evidence

Check necessity, dependencies, risk and capacity before selecting a delivery sequence. A higher optional-feature score does not erase a genuine mandatory commitment.

Option	Reach	Impact	Conf.	Effort	Score
A	100	1	0.80	2	40
B	300	0.5	0.60	3	30
C	80	2	0.90	2	72

TAKE IT TO YOUR PROJECT

Recalculate all three scores before discussing the order.

Test confidence and sensitivity

A ranking can be fragile when an estimate has weak support.

Change one assumption

If C's confidence falls from 0.90 to 0.40, its score becomes 80 x 2 x 0.40 / 2 = 32.

Read the effect

A remains 40 and B 30, so the order becomes A, C, B. The changed ranking shows why supporting evidence matters.

Choose a learning action

A small user test or technical investigation may be worth doing before a larger commitment if it resolves the uncertainty driving the choice.

TAKE IT TO YOUR PROJECT

Identify the estimate most likely to change your ranking.

Account for dependencies and overlap

Scores do not automatically provide an executable sequence.

Review the system

Check enabling work, shared resources, technical dependencies, user overlap and the effect of doing options together. Avoid summing reach as if every user were distinct.

App case

Cancellation needs a shared identity change outside the three scored options. Include that work and its timing before promising the cancellation release.

Make the trade-off visible

Record when a lower-ranked enabling item must come first or an option becomes less valuable after another is delivered.

TAKE IT TO YOUR PROJECT

Find one dependency not represented in a candidate's effort estimate.

Use the methods together

Necessity and comparative opportunity can support different parts of the decision.

A practical sequence

Clarify the objective and binding needs, define a viable scope with MoSCoW where appropriate, then use a consistent comparison such as RICE for suitable candidates.

App case

The release must support a valid booking. Among optional improvements, the team compares cancellation and reminders, checks the evidence and considers the identity dependency.

Avoid mechanical decisions

Do not force a numerical score to decide a matter it does not represent. Explain judgement, authority and residual uncertainty alongside the method.

TAKE IT TO YOUR PROJECT

State which question each method answers in your decision.

Record the chosen priority

A useful record explains why the team is committing to the work.

Capture the basis

Record the objective, period, option definitions, categories or scores, evidence, assumptions, dependencies, decision owner and selected sequence.

App case

The team keeps the valid-booking commitment, investigates the identity dependency and plans the next useful cancellation slice if the evidence supports it.

Communicate the outcome

Tell requesters what was selected, deferred or excluded and the reason. Link delivery work to the current decision so changes can be understood.

TAKE IT TO YOUR PROJECT

Write a short priority decision with the facts that would change it.

Review priorities from new evidence

An agreed order should remain useful as the project learns.

Use review triggers

Inspect material feedback, changed constraints, delivery results, estimates or strategy at suitable decision points. Keep current commitments and authority in view.

App case

A pilot shows reminders affect fewer users than expected. Update the reach assumption and examine the decision impact rather than preserving an obsolete score for consistency.

Keep history

Record why the order changed. Distinguish learning from constant unexamined interruption so the team can deliver coherent results.

TAKE IT TO YOUR PROJECT

Name a specific finding that would trigger a priority review.



Learn from 350+ top universities and companies

ILLINOIS

Duke UNIVERSITY

Google

UNIVERSITY OF MICHIGAN

IBM

VANDERBILT UNIVERSITY

JOHNS HOPKINS UNIVERSITY

Invest in your career

Explore new skills

Access 10,000+ courses in AI, business, technology, and more.

Your prioritisation desk check

Good prioritisation makes a feasible, explainable commitment.

Check the inputs

Objective, horizon, mandatory conditions, candidate scope, units and evidence should be clear. Challenge weak estimates and inconsistent scales.

Check the choice

Are dependencies, capacity, risks, overlap and authority addressed? Can the team explain why the selected work best serves the intended result?

Use the app case

Scores of 40, 30 and 72 support a conversation only under the stated assumptions. Combine them with necessity and delivery evidence before acting.

TAKE IT TO YOUR PROJECT

Review one ranked list and identify the missing decision factor.