

# Wiegers' Prioritization Matrix

## Step 1. List all of the requirements, features, or use cases that you wish to prioritize in a spreadsheet

We'll use features in this example. All of the items must be at the same level of abstraction. For example, don't mix individual requirements with product features. If certain features are logically linked (that is, you would only implement feature B if feature A were included as well), include only the driving feature in the analysis. This model will work with up to several dozen features before it becomes unwieldy. If you have more items than that, abstract related features together to create a more manageable initial list. You can do a second round of analysis at a finer granularity of requirements detail if you need to.

## Step 2. Estimate the relative benefit that each feature provides to the customer or the business on a scale from 1 to 9

..., with 1 indicating very little benefit and 9 being the maximum possible benefit. These benefits indicate alignment with the product's business requirements. Your customer representatives are the best people to judge these benefits.

## Step 3. Estimate the relative penalty the customer or business would suffer if the feature is not included

Again, use a scale from 1 to 9, where 1 means essentially no penalty and 9 indicates a very serious downside. For example, failing to comply with a government regulation could incur a high penalty even if the customer benefit is low, as would omitting a feature that any reasonable customer would expect, whether or not they explicitly requested it. Requirements that have both a low benefit and a low penalty add cost but little value; they may be instances of gold plating.

## Step 4. The Total Value column is the sum of the relative benefit and penalty

By default, benefit and penalty are weighted equally. As a refinement, you can change the weights for these two factors. In Table 2, all benefit ratings are weighted twice as heavily as the penalty ratings. The spreadsheet totals the feature values and calculates the percentage of the total product value provided by these features that is attributable to each feature.

## Step 5. Estimate the relative cost of implementing each feature

..., again on a scale ranging from a low of 1 to a high of 9. The spreadsheet will calculate the percentage of total cost for each feature. Developers estimate the cost ratings based on factors such as the requirement complexity, the extent of user interface work required, the potential ability to reuse existing designs or code, and the levels of testing and documentation needed.

## Step 6. Developers estimate the relative degree of technical or other risk associated with each feature

...on a scale from 1 to 9. An estimate of 1 means you can program it in your sleep, while 9 indicates serious concerns about feasibility, the availability of staff with the needed expertise, or the use of unproven or unfamiliar tools and technologies. The spreadsheet will calculate the percentage of the total risk that comes from each feature.

By default, cost and risk are weighted equally, and each carries the same weight as the benefit and penalty terms. You can adjust the cost and risk weightings in the spreadsheet. In Table 2, risk has one-half the weight of the cost factor, which has the same weight as the penalty term. If you don't want to consider risk in the model, set the risk weighting value to zero.

## Step 7. Once you enter the estimates into the spreadsheet, it calculates a priority number for each feature.

The formula for the Priority column is..... 
$$\text{Priority} = \frac{\text{Value}\%}{\text{Cost}\% * \text{CostWeight} + \text{Risk}\% * \text{RiskWeight}}$$

## Step 8. Sort the list of features in descending order by calculated priority.

The features at the top of the list have the most favourable balance of value, cost, and risk, and thus should have higher implementation priority. The key customer and developer representatives should review the completed spreadsheet to agree on the ratings and the resulting sequence. This semi-quantitative method is not mathematically rigorous, and it is limited by your ability to estimate the benefit, penalty, cost, and risk for each item. Therefore, use the calculated priority sequence only as a guideline. It will take you awhile to calibrate your rating scales for a set of requirements, so iterate through the list after rating all the requirements and make any necessary adjustments. Calibrate this model for your own use with a set of completed requirements or features from a previous product. Adjust the weighting factors until the calculate priority sequence agrees with your after-the-fact evaluation of how important the requirements in your test set really were.

This model can also help you make trade-off decisions when you're evaluating proposed new requirements. Estimate their priority using the model to tell you how they match up against existing requirements, so you can choose an appropriate implementation sequence. Any actions you can take to move requirements prioritization from the political arena into an objective and analytical one will improve the project's ability to deliver the most important functionality in the most appropriate sequence.