Requirements Engineering

2019/2020

Course 5

Course 5 outline

- Requirements Prioritization

Requirements Prioritization

- Prioritization is an important step towards making good decisions regarding product planning for single and multiple releases.
- Various aspects of functionality are considered, such as importance, risk, cost, etc.
- Prioritization decisions are made by stakeholders, including users, managers, developers, or their representatives.
- When developing software systems, multiple trade-offs must be made.
- The functionality that is most important for the customers might not be as important when other aspects (e.g. price) are factored in.
- The functionality that is most desired by the customers, as well as least risky, least costly, etc must be developed.
- Requirements prioritization is a fundamental activity for project success.

Requirements Prioritization

- Prioritization helps to identify the most valuable requirements from the set of requirements identified during elicitation by distinguishing the critical few from the trivial many.
- The process of prioritizing requirements provides support for different activities:
  - for stakeholders to decide on the core requirements for the system
  - to plan and select an ordered, optimal set of software requirements for implementation in successive releases
  - to trade off desired project scope against sometimes conflicting constraints such as schedule, budget, resources, time to market, and quality
  - to balance the business benefit of each requirement against its cost
  - to balance implications of requirements on the software architecture and future evolution of the product and its associated cost
  - to select only a subset of the requirements and still produce a system that will satisfy the customer(s)
  - to estimate expected customer satisfaction
  - to minimize rework and schedule slippage (plan stability)
  - to handle contradictory requirements, focus the negotiation process, and resolve disagreements between stakeholders
  - to establish relative importance of each requirement to provide the greatest value at the lowest cost

Requirements Prioritization

- The result of prioritization might form the basis of product and marketing plans, as well as being a driving force during project planning.
- “The challenge is to select the “right” requirements out of a given superset of candidate requirements so that all the different key interests, technical constraints and preferences of the critical stakeholders are fulfilled and the overall business value of the product is maximized”(Ruhe, Eberlein, Pfahl, Quantitative Win-Win)
- It is possible to rectify incorrect decisions later on via change management but this can be very costly since it is significantly more expensive to correct problems later in the development process.
- The most cost effective way of developing software is to find the optimal set of requirements early, and then to develop the software according to this set. To accomplish this, it is crucial to prioritize the requirements to enable selection of the optimal set.

Requirements Prioritization

- Prioritizing requirements can also have benefits (e.g. it is possible to find requirements defects (e.g misjudged, incorrect and ambiguous requirements) since requirements are analyzed from a perspective that is different from that taken during reviews of requirements).
- Prioritization techniques can be divided into two categories: methods and negotiation approaches.
- The methods are based on quantitatively assigning values to different aspects of requirements.
- The negotiation approaches focus on giving priorities to requirements by reaching agreement between different stakeholders.

Aspects of Prioritization

- Requirements can be prioritized taking many different aspects into account.
- An aspect is a property or attribute of a project and its requirements that can be used to prioritize requirements.
- Common aspects are importance, penalty, cost, time, and risk.
- Prioritizing requirements based on a single aspect - easy to decide which one is most desirable.
- Other aspects are involved, such as cost, customers can change their mind and high priority requirements may turn out to be less important if they are very expensive to satisfy.
- Often, the aspects interact and changes in one aspect could result in an impact on another aspect.

Aspects of Prioritization: Importance

- When prioritizing importance, the stakeholders should prioritize which requirements are most important for the system.
- Importance could be a multifaceted concept - it depends very much on which perspective the stakeholder has.
- Importance could be:
  - urgency of implementation,
  - importance of a requirement for the product architecture,
  - strategic importance for the company,
  - etc.
- It is essential to specify which kind of importance the stakeholders should prioritize in each case.

Aspects of Prioritization: Penalty

- It is possible to evaluate the penalty that is introduced if a requirement is not fulfilled.
- Penalty is not just the opposite of importance.
  - E.g., failing to conform to a standard could incur a high penalty even if it is of low importance for the customer (i.e. the customer does not get excited if the requirement is fulfilled).
  - Same for implicit requirements that users take for granted, and whose absence could make the product unsuitable for the market.

Aspects of Prioritization: Cost & Time

- The implementation cost is usually estimated by the developing organization.
- Measures that influence cost include: complexity of the requirement, the ability to reuse existing code, the amount of testing and documentation needed, etc.
- Cost is often expressed in terms of staff hours (effort) since the main cost in software development is often primarily related to the number of hours spent.
- Time (i.e. lead time) is influenced by many other factors such as degree of parallelism in development, training needs, need to develop support infrastructure, complete industry standards, etc.

- Cost and time could be prioritized by using any of the prioritization techniques, but also by simply estimating the actual cost on an absolute or normalized scale.

Aspects of Prioritization: Risk

- Every project carries some amount of risk.
- In project management, risk management is used to cope with both internal (technical and market risks) and external risks (e.g. regulations, suppliers).
- Both likelihood and impact must be considered when determining the level of risk of an item or activity.
- Risk management can also be used when planning requirements into products and releases by identifying risks that are likely to cause difficulties during development.
- Such risks could include performance risks, process risks, schedule risks etc.
- It is possible to calculate the risk level of a project based on the estimated risk likelihood and risk impact for each requirement.

Aspects of Prioritization: Volatility

- Volatility of requirements is considered a risk factor and is sometimes handled as part of the risk aspect.
- Others think that volatility should be analyzed separately and that volatility of requirements should be taken into account separately in the prioritization process.
- The reasons for requirements volatility vary. E.g. the market changes, business requirements change, legislative changes occur, users change, or requirements become clearer during the software life cycle.
- Volatile requirements affect the stability and planning of a project, and presumably increase the costs since changes during development increase the cost of a project.
- Further, the cost of a project might increase because developers have to select an architecture suited to change if volatility is known to be an issue.

Aspects of Prioritization: Other aspects

- Examples of other aspects are:
  - financial benefit,
  - strategic benefit,
  - competitors, competence / resources,
  - release time,
  - ability to sell.
- The stakeholders should develop a list of important aspects to use in the decision-making.
- It is important that the stakeholders have the same interpretation of the aspects as well as of the requirements.
- Studies have shown that it is hard to interpret the results if no guidelines about the true meaning of an aspect are present.

Combining Different Aspects

- It is important to consider multiple aspects before deciding if a requirement should be implemented directly, later, or not at all.
- For example, in the Cost-Value approach, both value (importance) and cost are prioritized to implement those requirements that give most value for the money.
- The Planning Game (PG) from Extreme Programming uses a similar approach when importance, effort (cost), and risks are prioritized.
- Importance and stability (volatility) are also suggested as aspects that should be used when prioritizing while others suggest that dependencies also must be considered.
- Which aspects to consider depends very much on the specific situation.
- It is important to know about possible aspects and how to combine them efficiently for the specific project.

Prioritization Techniques

- The purpose of any prioritization is to assign values to distinct prioritization objects that allow establishment of a relative order between the objects in the set.
- In RE, the objects to be prioritized are the requirements.
- The prioritization can be done with various measurement scales and types.
- The least powerful prioritization scale is the ordinal scale, where the requirements are ordered so that it is possible to see which requirements are more important than others, but not how much more important.
- The ratio scale is more powerful since it is possible to quantify how much more important one requirement is than another (the scale often ranges from 0 - 100 percent).
- An even more powerful scale is the absolute scale, which can be used in situations where an absolute number can be assigned (e.g. number of hours).
- Some techniques assume that each requirement is associated with a priority, and others group requirements by priority level.

Analytical Hierarchy Process (AHP)

- The Analytic Hierarchy Process (AHP) is a systematic decision-making method that has been adapted for prioritization of software requirements.
- It is conducted by comparing all possible pairs of hierarchically classified requirements, in order to determine which has higher priority, and to what extent (usually on a scale from one to nine where one represents equal importance and nine represents absolutely more important).
- The total number of comparisons to perform with AHP are n × (n-1)/2 (where n is the number of requirements) at each hierarchy level.
- Studies have shown that AHP is not suitable for large numbers of requirements.
- Variants of the technique have been found to reduce the number of comparisons by as much as 75 percent.
- The result from a prioritization with AHP is a weighted list on a ratio scale.
- AHP example: http://mat.gsia.cmu.edu/classes/mstc/multiple/node4.html

Cumulative Voting, the 100-Dollar Test

- The 100-dollar test is a very straightforward prioritization technique where the stakeholders are given 100 imaginary units (money, hours, etc.) to distribute between the requirements.
- The result of the prioritization is presented on a ratio scale.
- A problem with this technique arises when there are too many requirements to prioritize. For example, if you have 25 requirements, there are on average four points to distribute for each requirement.
- Another possible problem with the 100-dollar test (especially when there are many requirements) is that the person performing the prioritization miscalculates and the points do not add up to 100.
- This technique should be performed only once on the same set of requirements, since the stakeholders might bias their evaluation the second time around if they do not get one of their favorite requirements as a top priority.
- Similarly, some clever stakeholders might put all their money on a favorite requirement that others do not prioritize as highly, while not giving money to requirements that will get much money anyway (e.g. response time).

Numerical Assignment (Grouping)

- Numerical assignment is the most common prioritization technique and is suggested both in RFC 2119 and IEEE Std. 830-1998.
- The approach is based on grouping requirements into different priority groups. The number of groups can vary, but in practice, three groups are very common.
- It is important that each group represents something that the stakeholders can relate to (e.g. critical, standard, optional), for a reliable classification.
- Using relative terms such as high, medium, and low will confuse the stakeholders. (Important when there are stakeholders with different views of what high, medium and low means.) A clear definition of what a group really means minimizes such problems.
- Another problem: stakeholders tend to think that everything is critical.
  - Eg. customers prioritize using three groups; critical, standard, and optional, they will most likely consider 85% as critical, 10% as standard, and 5% as optional. One idea is to put restrictions on the allowed number of requirements in each group (e.g. not less than 25 % of the requirements in each group)
- The result is requirements prioritized on an ordinal scale. However, the requirements in each group have the same priority, which means that each requirement does not get a unique priority.

Ranking

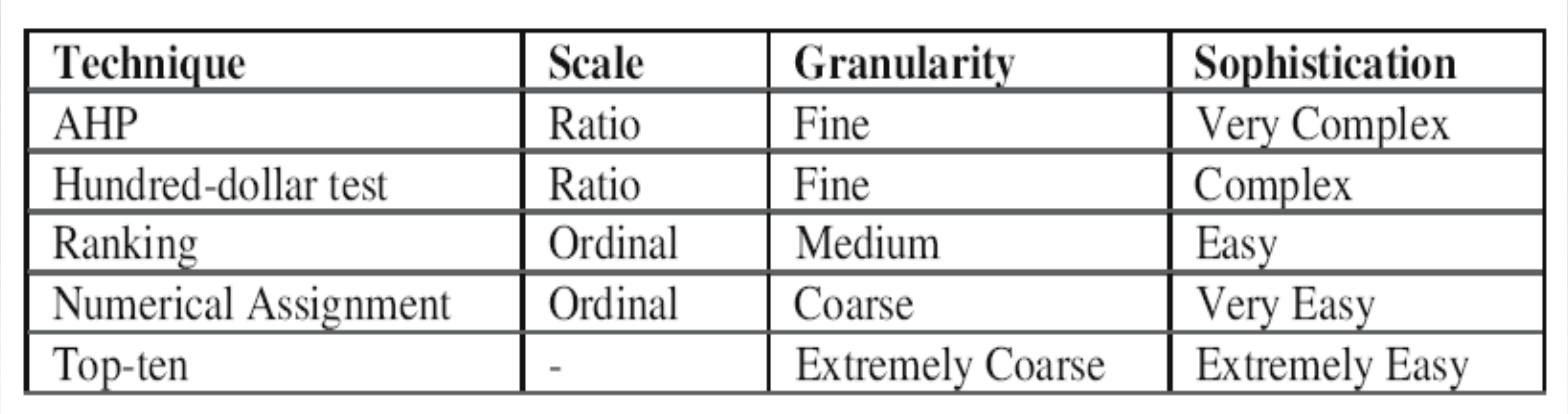
- Ranking is also based on an ordinal scale but the requirements are ranked without ties in rank.
- The most important requirement is ranked 1 and the least important is ranked n (for n requirements).
- Each requirement has a unique rank (in comparison to numerical assignment) but it is not possible to see the relative difference between the ranked items (as in AHP or the 100-dollar test).
- The list of ranked requirements could be obtained in a variety of ways (bubble sort or binary search tree algorithms, etc)
- Ranking seems to be more suitable for a single stakeholder because it might be difficult to align several different stakeholders' views.
- It is possible to combine the different views by taking the mean priority of each requirement but this might result in ties for requirements which this method wants to avoid.

Top-Ten Requirements

- The stakeholders pick their top-ten requirements (from a larger set) without assigning an internal order between the requirements.
- It makes the approach especially suitable for multiple stakeholders of equal importance.
- The reason to not prioritize further is that it might create unnecessary conflict when some stakeholders get support for their top priority and others only for their third priority.
- Conflicts might arise anyway if, for example, one customer gets three top-ten requirements into the product while another gets six top-ten requirements into the product.
- It is important to not just take an average across all stakeholders since it might lead to some stakeholders not getting any of their top requirements.
- It is crucial that some essential requirements are satisfied for each stakeholder. This could obviously result in a situation that dissatisfies all customers instead of satisfying a few customers completely.
- The main challenge in this technique is to balance these issues.

Which Technique to Choose?

- A general advice is to use the simplest appropriate prioritization technique and use more sophisticated ones when a more sensitive analysis is needed for resolving disagreements or to support the most critical decisions.
- More sophisticated techniques generally are more time consuming, the simplest possible technique ensures cost effective decisions.



Technique	Scale	Granularity	Sophistication
AHP	Ratio	Fine	Very Complex
Hundred-dollar test	Ratio	Fine	Complex
Ranking	Ordinal	Medium	Easy
Numerical Assignment	Ordinal	Coarse	Very Easy
Top-ten	-	Extremely Coarse	Extremely Easy

Combining Techniques

- It is possible to combine some of them to make prioritization easier or more efficient.
- Some combinations of the techniques already exist. Eg. Planning Game (PG) in Extreme Programming.
- In requirements prioritization, there are requirements that must be in the product (e.g. platform requirements), requirements that the product clearly need not satisfy (e.g. very optional requirements), and requirements that need more attention.
- This means that the requirements are assigned to one of three groups (numerical assignment) and requirements that need more attention are prioritized by any of the other techniques (AHP, ranking, 100 points etc.). In this approach, not all requirements must be prioritized by a more sophisticated technique, which decreases the effort.
- Which method or combination of methods is suitable often depends on the individual project.

Stakeholders and Prioritization

- Since requirements can be prioritized from several different aspects, different roles must also be involved in the prioritization process to get the correct views (e.g. product managers prioritize strategic importance and project managers prioritize risks).
- At least three perspectives should always be represented: customers, developers, and financial representatives.
- Each of these stakeholders provides vital information that the other two may neglect or are unable to produce since customers care about the user/customer value, developers know about the technical difficulties, and financial representatives know and care for budgetary constraints and risks.
- It is suitable to involve all perspectives (beside these three) that have an interest in the project or product.

Reprioritization

- It is likely that new requirements will arrive, requirements are deleted, priorities of existing requirements change, or that the requirements themselves change.
- It is very important that the prioritization process is able to deal with changing requirements and priorities of already prioritized requirements.
- When prioritizations are on an ordinal (e.g. ranking and numerical assignment) or absolute scale (estimating cost) this does not introduce any major problems since the new or changed requirement just need to be assigned a value, or a correct priority.
- When using prioritization on a ratio scale (such as AHP), the situation becomes more complex since all requirements should be compared to all others to establish the correct relative priorities. However, it is possible to tailor this process by comparing new or modified requirements with certain reference requirements and so estimating the relative value.

Non-Functional Requirements

- It is not always advisable to prioritize functional and non-functional requirements together.
- Differences include:
  - Functional requirements usually relate to specific functions while nonfunctional requirements usually affect several functions (from a collection of functions to the whole system).
  - Non-functional requirements are properties that the functions or system must have, implying that non-functional requirements are useless without functional requirements.
  - When implemented, functional requirements either work or not while nonfunctional requirements often have a “sliding value scale” of good and bad.
- It is possible to prioritize them separately with the same or even with different techniques. Some techniques are especially suitable for prioritizing non-functional requirements (e.g. conjoint analysis).

Evaluating Prioritization

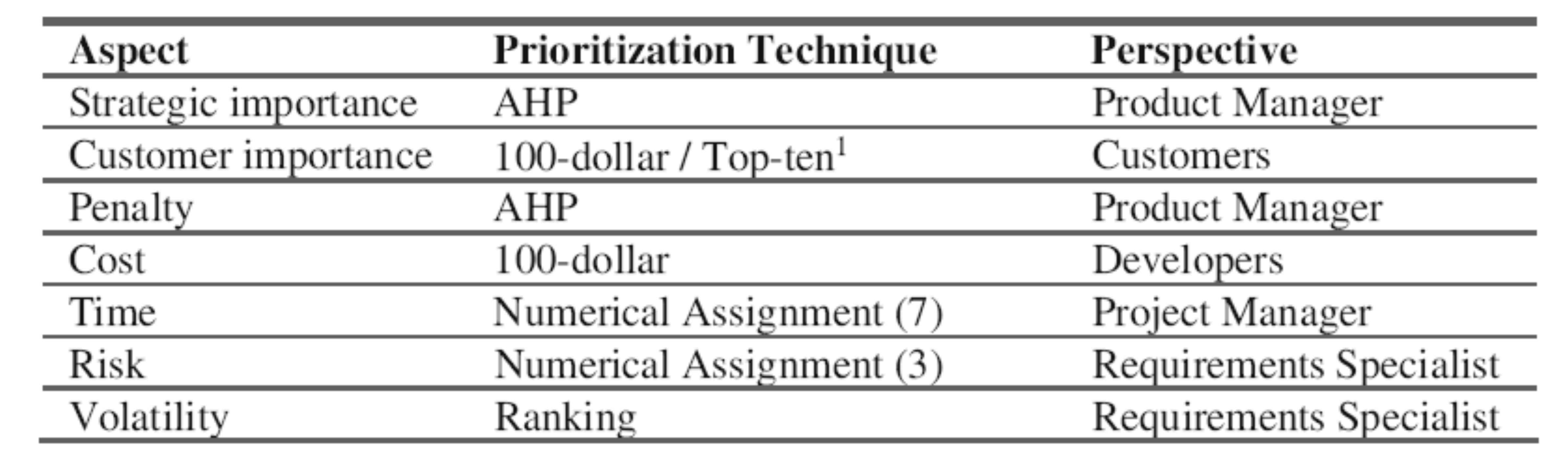
- It is necessary to evaluate the result of prioritizations in retrospect (for improving and/or adjusting).
- It is important that information about the priorities is kept since these provide the best information for analyzing both the product and the process (both selected and discarded requirements from a release).
- It is possible to do post analysis to evaluate if the correct requirements were selected and if they fulfilled the stakeholders' expectations.
- If they did not, it is possible to change the process and the product for subsequent products/releases to get better prioritizations and more satisfied stakeholders.
- One way of evaluating if the correct priorities were assigned is through gap-analysis where the "gap" between perceived levels of fulfillment of a requirement and the importance of the requirement is calculated.

Using the Results of Prioritization

- The results of a prioritization exercise must be used carefully.
- Dependencies between requirements should be taken into consideration when choosing which requirements to include.
  - Dependencies could be related to cost, value, changes, people, competence, technical precedence, etc.
  - Such dependencies might force one requirement to be implemented before another, implying that it is not possible to just follow the prioritization list.
- The product may have some naturally built-in constraints.
  - For example, projects have constraints when it comes to effort, quality, duration, etc.
  - Such constraints makes the selection of which requirements to include in a product more complex than if the choice were solely based on the importance of each requirement.
  - A common approach to make this selection is to propose a number of alternative solutions from which the stakeholders can choose the one that is most suitable based on all implicit context factors.

Example of a Requirements Prioritization

- The example analyzes 15 requirements (R1-R15) in a situation with three known customers.
- Each of the 15 requirements is prioritized according to the different aspects:



Aspect	Prioritization Technique	Perspective
Strategic importance	AHP	Product Manager
Customer importance	100-dollar / Top-ten[1]	Customers
Penalty	AHP	Product Manager
Cost	100-dollar	Developers
Time	Numerical Assignment (7)	Project Manager
Risk	Numerical Assignment (3)	Requirements Specialist
Volatility	Ranking	Requirements Specialist

- Remarks:
  - Numerical assignment for time (7) and risk (3) uses a different number of groups to show varying levels of granularity.
  - The customer importance is prioritized both by the top-ten technique and the 100-dollar technique depending how much time and cost the different customers consider reasonable.

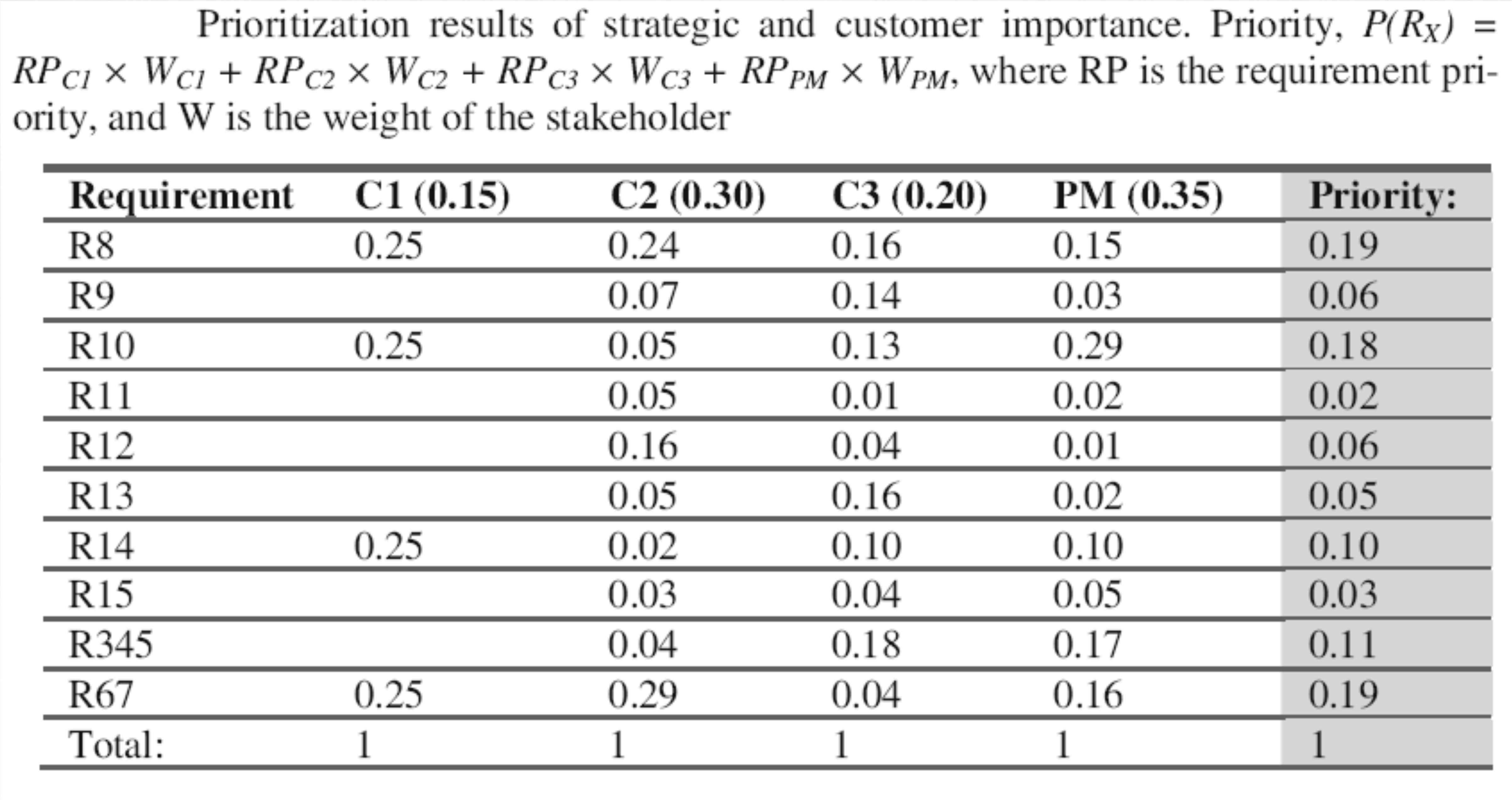
Example of a Requirements Prioritization

Remarks:

- Requirements R1 and R2 are requirements that are absolutely necessary to get the system to work at all. They are not prioritized by the customers but they are estimated when it comes to cost, risk, etc. since R1 and R2 influence these variables no matter what.
- Two groups of requirements have been identified as having high dependencies (must be implemented together) and should be prioritized together. Requirements R3, R4, and R5 are grouped together as R345, and requirements R6 and R7 are grouped into R67.

Example of a Requirements Prioritization

1. Prioritize the importance of the requirements (performed by three known customers and the product manager, C1 used the top-ten technique)

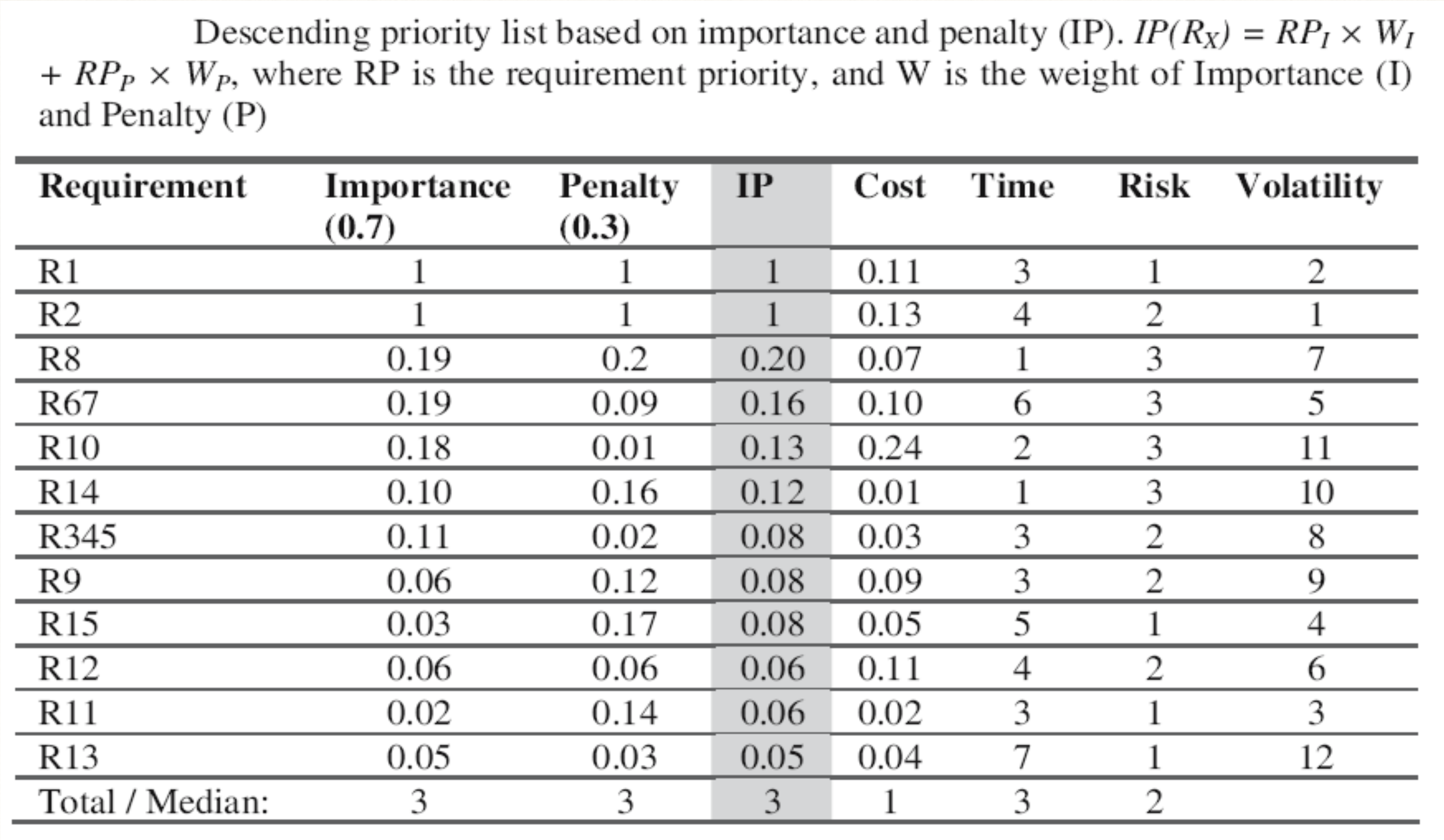


Prioritization results of strategic and customer importance. Priority, $P(R_X) = RP_{C1} \times W_{C1} + RP_{C2} \times W_{C2} + RP_{C3} \times W_{C3} + RP_{PM} \times W_{PM}$, where RP is the requirement priority, and W is the weight of the stakeholder

Requirement	C1 (0.15)	C2 (0.30)	C3 (0.20)	PM (0.35)	Priority:
R8	0.25	0.24	0.16	0.15	0.19
R9		0.07	0.14	0.03	0.06
R10	0.25	0.05	0.13	0.29	0.18
R11		0.05	0.01	0.02	0.02
R12		0.16	0.04	0.01	0.06
R13		0.05	0.16	0.02	0.05
R14	0.25	0.02	0.10	0.10	0.10
R15		0.03	0.04	0.05	0.03
R345		0.04	0.18	0.17	0.11
R67	0.25	0.29	0.04	0.16	0.19
Total:	1	1	1	1	1

Example of a Requirements Prioritization

2. The next step is to prioritize based on the other aspects. In this case, the Priority from step 1 is used to express Importance. The requirements R1 and R2 (absolutely necessary) have been added.



Descending priority list based on importance and penalty (IP). $IP(R_X) = RP_I \times W_I + RP_P \times W_P$, where RP is the requirement priority, and W is the weight of Importance (I) and Penalty (P)

Requirement	Importance (0.7)	Penalty (0.3)	IP	Cost	Time	Risk	Volatility
R1	1	1	1	0.11	3	1	2
R2	1	1	1	0.13	4	2	1
R8	0.19	0.2	0.20	0.07	1	3	7
R67	0.19	0.09	0.16	0.10	6	3	5
R10	0.18	0.01	0.13	0.24	2	3	11
R14	0.10	0.16	0.12	0.01	1	3	10
R345	0.11	0.02	0.08	0.03	3	2	8
R9	0.06	0.12	0.08	0.09	3	2	9
R15	0.03	0.17	0.08	0.05	5	1	4
R12	0.06	0.06	0.06	0.11	4	2	6
R11	0.02	0.14	0.06	0.02	3	1	3
R13	0.05	0.03	0.05	0.04	7	1	12
Total / Median:	3	3	3	1	3	2	

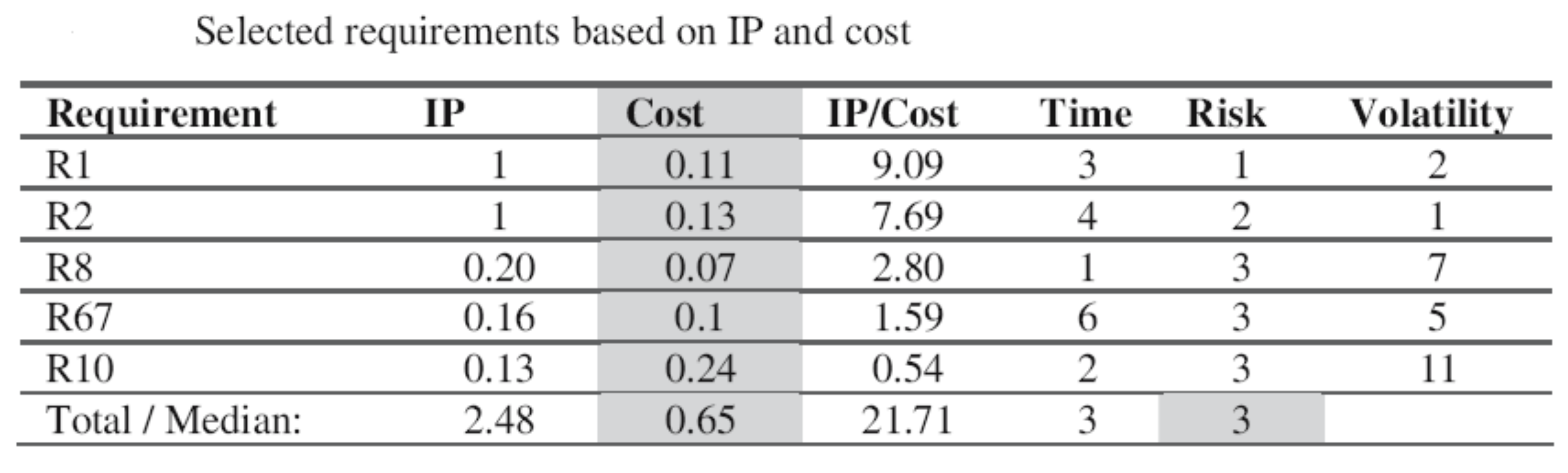
Example of a Requirements Prioritization

- With this information there are two options:
  1. pick prioritized items from the top of the list until the cost constraints are reached
  2. analyze further based on other prioritized aspects, if prioritizations of additional aspects are available.
- The example has two major constraints:
  - the iteration is not allowed to cost more than 65% of the total cost of the elicited requirements,
  - the median risk level of the requirements included is not allowed to be higher than 2.5.

Example of a Requirements Prioritization

Solution 1: to include the requirements with the highest IP.

The list was cut when the sum of costs reached 65% of the total cost of elicited requirements.



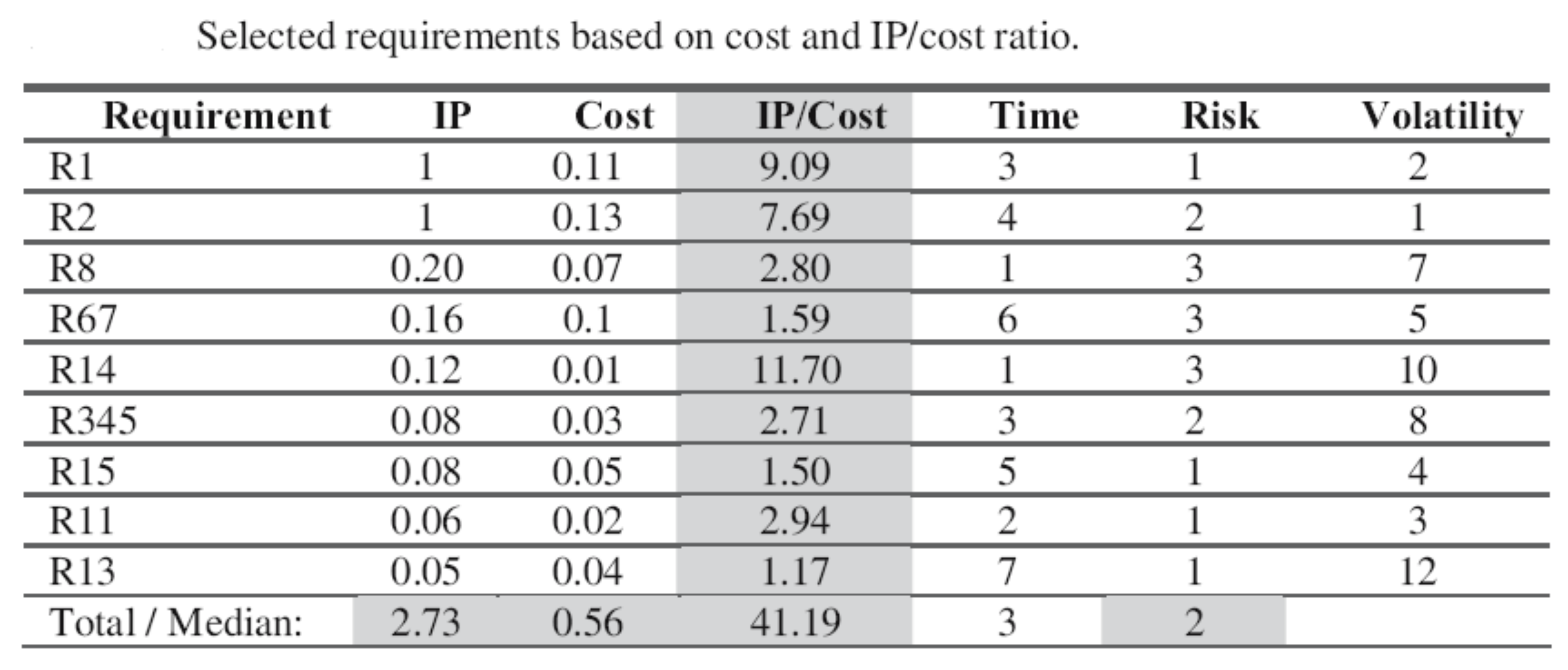
Selected requirements based on IP and cost

Requirement	IP	Cost	IP/Cost	Time	Risk	Volatility
R1	1	0.11	9.09	3	1	2
R2	1	0.13	7.69	4	2	1
R8	0.20	0.07	2.80	1	3	7
R67	0.16	0.1	1.59	6	3	5
R10	0.13	0.24	0.54	2	3	11
Total / Median:	2.48	0.65	21.71	3	3	

- Fits within the cost constraints but could not satisfy the risk constraint. As a result, the project becomes too risky.

Example of a Requirements Prioritization

Solution 2: take the IP/Cost ratio into consideration. This shows which requirements provide most IP at the least cost. A limit of only selecting requirements that have an IP/Cost-ratio higher than 1.0 was set up.



Selected requirements based on cost and IP/cost ratio.

Requirement	IP	Cost	IP/Cost	Time	Risk	Volatility
R1	1	0.11	9.09	3	1	2
R2	1	0.13	7.69	4	2	1
R8	0.20	0.07	2.80	1	3	7
R67	0.16	0.1	1.59	6	3	5
R14	0.12	0.01	11.70	1	3	10
R345	0.08	0.03	2.71	3	2	8
R15	0.08	0.05	1.50	5	1	4
R11	0.06	0.02	2.94	2	1	3
R13	0.05	0.04	1.17	7	1	12
Total / Median:	2.73	0.56	41.19	3	2	

Example of a Requirements Prioritization

Solution 2:

- The cost constraints are still met (even nine percent less cost) while also satisfying the risk constraint.
- the IP-value of the second candidate solution is higher which indicates that the customers are more satisfied with the product and the IP/Cost ratio is almost doubled.
- The second candidate solution satisfies 91 percent (2.73/3) of the IP aspect, compared to 83 percent in the first candidate solution.
- The fact that the second solution costs less and is less risky favors solution 2.

Volere Prioritization Analysis

Prioritization Factors- factors that commonly affect prioritization decisions are some combination of:

- **Minimize Cost** of implementation (how much cost to develop?)
- **Value** to **customer** (how much does the customer want it?)
- **Time** to **implement** (how much time to deliver?)
- **Ease** of **technical** implementation (how technologically difficult?)
- **Ease** of **business** implementation (how organizationally difficult?)
- **Value** to the **business** (how much will the business benefit?)
- **Obligation** to some **external authority** (necessity to obey law?)

www.volere.co.uk