# 7 Usage Scenarios of a CoBRA Model

The CoBRA method has been designed to provide a project decision maker with comprehensive support regarding estimating, controlling, and managing project effort. The CoBRA model can be used for a number of software estimation purposes.

In this chapter, we present several typical scenarios of using the CoBRA model for different purposes. For each scenario we explain, using an intuitive example, how to interpret appropriately the outcomes of applying the CoBRA model.

## 7.1 Effort Estimation

### 7.1.1 Most Likely Effort

Traditionally, the objective of effort estimation has been to evaluate the most likely effort required to successfully complete a project with certain characteristics. The simplest way to obtain an estimate of the most likely project effort is to take the mean value from the distribution of effort provided as output by the CoBRA model (Fig. 7.1).

The CoBRA model consists of a quantified effort overhead model and a *baseline nominal productivity* determined using a set of historical projects. After feeding the effort overhead with actual factor data from the project, it returns a distribution of the project's *effort overhead* (distribution of relative frequency obtained through Monte Carlo simulation). Since usually not all project characteristics are known at the time of effort estimation, some of them may be estimated first and updated later, when project re-estimation is performed.

Effort overhead distribution and baseline nominal productivity are inputs to the basic CoBRA equation (7.1), which we use to estimate the effort required to deliver software products of a particular *size*.

A. Trendowicz, *Software Cost Estimation, Benchmarking, and Risk Assessment*,
The Fraunhofer IESE Series on Software and Systems Engineering,
DOI 10.1007/978-3-642-30764-5_7, © Springer-Verlag Berlin Heidelberg 2013

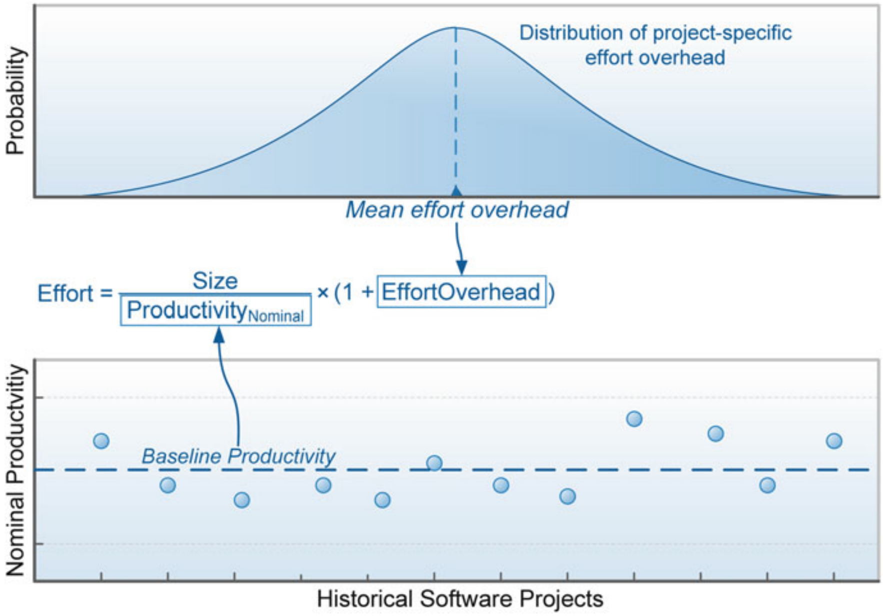


**Fig. 7.1** Estimating the most likely effort in CoBRA

$$Effort_{Act} = \frac{Size}{Productivity_{Nom}} \cdot (EO_{Act} + 1) \tag{7.1}$$

Let us consider the example distribution of estimated effort in Fig. 7.2. We adapt the mean value over the distribution as the estimate of the most likely project effort. The field under the distribution curve to the left of the mean point represents the probability of project effort being lower than the mean, whereas the field under the curve to the right of the mean represents the probability of project effort being larger than the mean. The latter probability is especially interesting from the point of view of project risk management because, in practice, it represents the probability of exceeding the estimated effort.

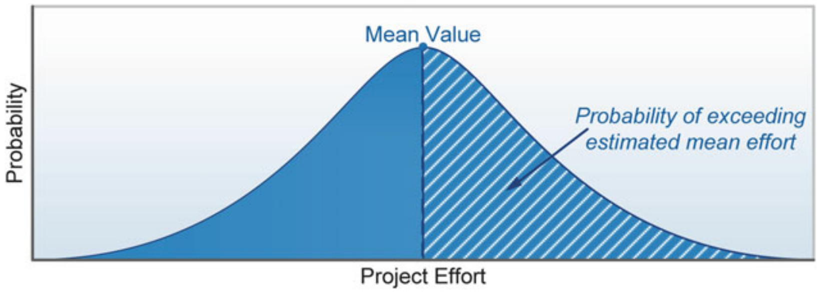


**Fig. 7.2** Example distribution of estimated project effort

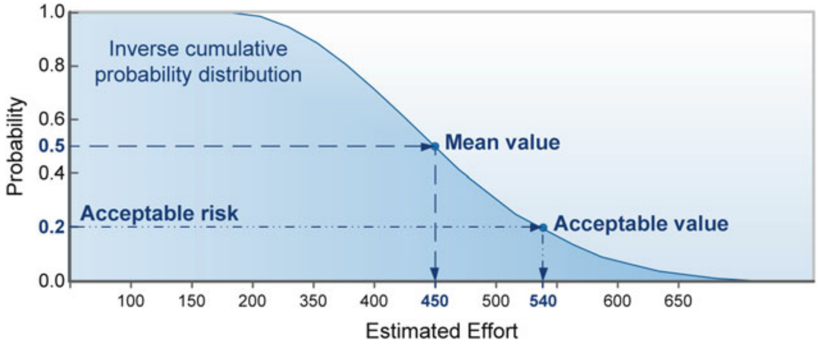


**Fig. 7.3** Example cumulative distribution of effort overhead

### 7.1.2 Effort at a Given Risk Level

In practice, we should avoid estimating project effort by simply taking the mean value over the estimated effort distribution. Instead, we should consider the risk[1] we are willing to take and estimate the effort for a specific probability of exceeding the estimated effort.

Let us consider a simple example. Figure 7.3 illustrates an inverse cumulative distribution of estimated project effort. We use this form of distribution because it is easier to interpret visually.

The most likely mean effort of 450 units means in practice that there is a 50 % chance of exceeding it. If we want to decrease the probability (risk) of running over the planned project budget, we must plan more. If we want to decrease the chance of exceeding the planned budget down to 20 %, we must plan 540 units of effort. In other words, in order to decrease the risk of exceeding the most likely estimate down to 0.2, one has to plan 20 % more effort.

## 7.2 Risk Management

The CoBRA method supports project risk management with respect to two aspects: (1) it handles the inherent uncertainty of software prediction and (2) it supports the identification of the potential sources of the most critical project risks related to development productivity and effort.

[1] In risk-driven estimation with CoBRA, we use the term risk as a synonym for probability. In the risk management domain, risk is defined as the composition of two elements: the probability of an event and the size of (negative) effects of the event.

**Managing Estimation Uncertainty** Considering uncertainty is an important element of managing risk while planning a software project. Although uncertainty is inherent to the prediction of software effort, estimation methods typically do not handle it properly, if at all. The objective of uncertainty management is to explicitly identify and consider the uncertainty of the input information on which the estimates are based as well as the uncertainty of the estimate itself. Identifying and understanding the sources of estimation uncertainty allows software managers to better handle prediction-related project risks and improve project budgeting and planning processes.

CoBRA supports the handling of estimation uncertainty with several mechanisms. First, the impact of each identified effort factor on effort is quantified using the three values representing a triangular probability distribution. Second, the actual value of each effort factor can be quantified by several values with an associated probability of occurrence. Finally, both the impact and the value of each effort factor can (and should) be quantified by multiple domain experts. Such uncertain input data are subject to a simulation algorithm that provides a probability distribution of estimated effort as its output.

**Managing Project Risks** The objective of risk management is to determine whether special actions are necessary to reduce effort-related risks in the project. CoBRA supports this objective in two ways:

1. It supports finding out how risky, with respect to effort, the project is going to be. This step consists of setting up a risk baseline of acceptable risk and assessing the risk with respect to this predefined baseline.
2. If the identified level of risk is already unacceptable, then CoBRA supports deciding on actions that should be undertaken to mitigate the risk.

## 7.2.1 Defining a Baseline for Risk Assessment

In order to perform a risk assessment, we have to build a baseline against which to evaluate individual software projects. We build such a baseline using data from a set of projects, each considered successful according to the organization's understanding of business success. The notion of a successful project may, for instance, at least encompass that the project was completed and did not create substantial financial loss. In cases where no additional project data is available, the same projects that were used for developing the CoBRA effort model can be utilized for setting up a risk baseline.

In CoBRA, we typically define a risk baseline as the median or the mean effort overhead upon the sample of successful historical projects. We explain this approach on an example presented in Fig. 7.4. The threshold $T_1$ represents the

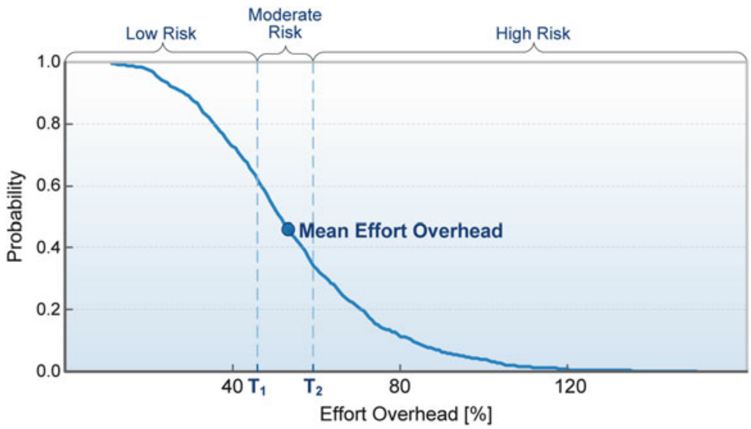


**Fig. 7.4** Simple risk assessment using mean effort overhead

median over the effort overheads of a set of historical projects, where for each historical project, the mean effort overhead over the distribution is considered. We may say that $T_1$ represents a "typical" project. This implies that 50 % of the projects will have a mean effort overhead value greater than $T_1$ and 50 % will have a mean value up to $T_1$.

An alternative formulation is to consider the probability of having an effort overhead exceeding the "majority" of projects. For a given sample of historical projects, we can additionally set up a threshold $T_2$ as the upper quartile upon the mean effort overheads of the historical projects. We may say that $T_2$ represents the "majority" of the projects. The upper quartile has 75 % of the projects below it and 25 % of the projects above it.

A comparison of a project's actual mean effort overhead against a baseline tells us how risky the project is. There are a number of different ways in which this can be done. In the following paragraphs, we present several approaches, starting from the simplest and progressing to the more complex ones.

## 7.2.2 Assessing Project Risk Level

The objective of project risk assessment is to evaluate how risky a project is. There are several strategies proposed in CoBRA for assessing the risk level of the software development of a project: based on effort overhead thresholds, based on an acceptable risk probability level, and based on an acceptable risk exposure level.

### Risk Assessment Based on Effort Overhead Thresholds

The first approach is based on determining effort overhead thresholds for a project. The thresholds delimit effort overhead intervals, which are judged more or less risky. Consequently, these intervals can be regarded as risk levels. We explained the idea of setting up risk thresholds in the previous paragraph and illustrated it in Fig. 7.4. Using this information, we can determine the risk level for the project.

The risk level for a project is defined as the interval into which its mean effort overhead falls with the mean effort overhead being calculated from the relative frequency distribution produced by the CoBRA model in a Monte Carlo simulation. Figure 7.4 illustrates two example thresholds $T_1$ and $T_2$. Based on these thresholds and on the project's mean effort overhead (*EO*), we can assess the risk of the project relative to already completed successful projects. We do this in the following way:

- If $EO < T_1$, the project falls into the group of low-risk projects.
- If $T_1 \leq EO \leq T_2$, the project falls into the group of moderate-risk projects.
- If $EO > T_2$, the project's falls into the group of high-risk projects.

Since the mean effort overhead of the example project in Fig. 7.4 falls between $T_1$ and $T_2$, it would be regarded as being of moderate risk. After the risk probability level for the project is determined, the preventive/corrective actions associated with that risk level are performed.

In CoBRA, we would typically set up the effort overhead thresholds $T_1$ and $T_2$ based upon the 50th and 75th percentiles,[2] respectively, from a sample of successful historical projects. In this case, half of the considered historical projects would have a mean effort overhead lower than $T_1$ (the 50th percentile), and 75 % of the historical projects would have an effort overhead that is lower than $T_2$ (the 75th percentile).[3] For reasons of convenience, we refer to these thresholds as representing the "typical" projects and the "majority" of the projects, respectively. These two thresholds define three risk levels: low, moderate, and high risk.

In general, we may define any reasonable number *n* of thresholds as *percentiles* upon the distribution of mean effort overhead of successful historical projects. A reasonable number would be between 1 and 5. When *n* thresholds are selected, then $n + 1$ risk levels have to be managed in the sense that for each risk level, specific actions have to be specified. In practice, the number and percentile values of the thresholds should be determined by experienced project managers in conjunction with quality assurance staff. The thresholds should be updated regularly as new and different types of projects are completed and as experience is gained in their use. A specific set of actions should be associated with each interval, except for the lowest one. The higher the risk probability level, the more consequential and costly these actions are likely to be due to the higher number and greater complexity of the software development processes that need to be addressed.

Figure 7.5 shows the two example thresholds derived from a sample of past projects and the curves of the cumulative effort overhead distribution for three

[2] The 25th, 50th, and 75th percentiles are referred to as lower, middle, and upper quartile. The middle quartile represents the median.

[3] These percentages may seem high, but it should be remembered that nominal projects never occur in practice and would consume very low and, at any rate, unrealistic effort, hence the large effort overhead percentages.

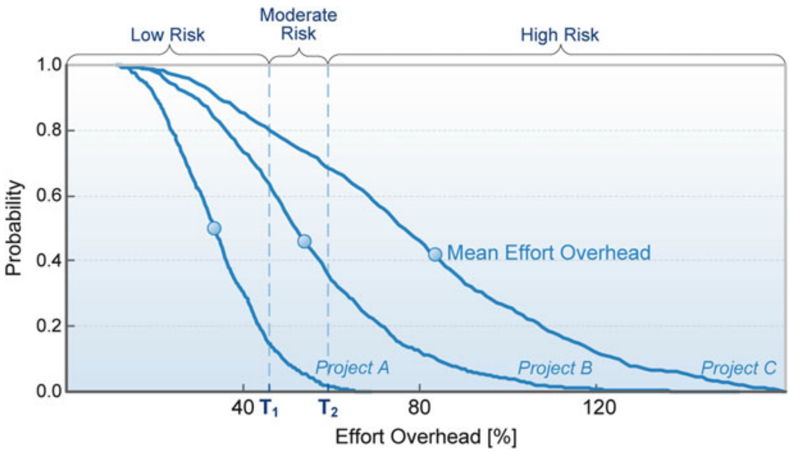


**Fig. 7.5** Example risk assessment based on risk threshold

hypothetical projects. For this example, the mean value of the effort overhead for *Project C* falls into the high-risk projects. Moreover, it has a probability of about 0.7 of exceeding $T_2$, that is, the effort overhead of the majority of the projects. *Project A*, on the other hand, is located in the low-risk class of projects and has a probability of merely 0.15 of exceeding $T_1$, which is the effort overhead of typical projects. Finally, *Project B* can be considered as a moderate-risk project compared to the already completed projects upon which the risk baseline (using thresholds) is based.

### Risk Assessment Based on Acceptable Risk Probability

The aforementioned simple, threshold-based approach is appealing because of its simplicity. However, it does not take into account the probability of the project's overhead falling into a different risk level. For example, if the mean effort overhead for a project falls in between $T_1$ and $T_2$—as *Project B* in Fig. 7.5—the project may still have a high probability of having an effort overhead exceeding $T_2$. Using the simple approach in our example, we would have designated *Project B* as having moderate risk, when in fact it still has a high probability (~0.35) of falling into the class of highly risky projects, as illustrated in Fig. 7.6. This means that the project manager performs the risk reduction actions for a moderate-risk project, whereas she/he should rather consider performing the risk reduction actions for a high-risk project.

To address this shortcoming, we must first define the concept of *Acceptable Risk Probability Level* (*ARP*). To continue from the example above, the acceptable risk probability answers the question of "how high does the probability of the effort overhead exceeding the 'majority' threshold have to be before we consider the project at high risk (instead of moderate risk)?" Therefore, we define acceptable risk probability as the maximum risk that the organization is willing to tolerate without taking actions to manage and reduce it.

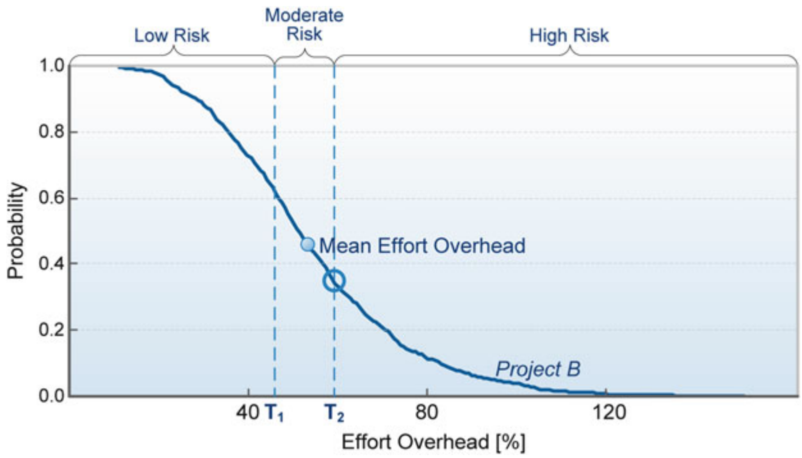


**Fig. 7.6** Threat of threshold-based risk assessment

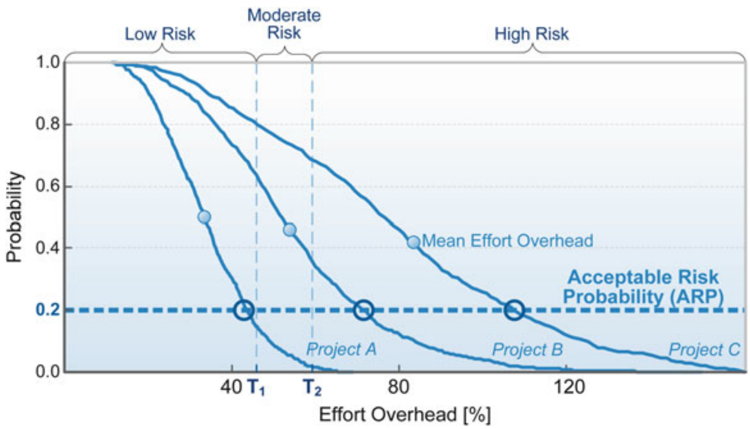


**Fig. 7.7** Example of risk assessment based on acceptable risk level

One way to specify acceptable risk is to set up a probability value that indicates the probability above which actions should be triggered. Continuing our example, in Fig. 7.7 we selected the value of $ARP = 0.2$ as the acceptable risk probability value for both the typical ($T_1$) and the majority ($T_2$) thresholds.

For the project for which risk assessment is to be performed, we run the effort overhead estimation model and generate the cumulative probability curve, like the curves for projects $A$, $B$, and $C$ in Fig. 7.7. In order to determine the risk level for a given project, we follow the effort overhead curve from left to right until we reach the intersection with the acceptable risk probability value that was chosen (bold dashed line at probability level 0.2 in Fig. 7.7). This intersection falls into one of the risk levels, and this is the risk level assigned to the project. In the example illustrated in Fig. 7.7, we show the cumulative effort overhead probability functions for three hypothetical projects, each of them belonging to one of the three risk levels defined by the two thresholds and the acceptable risk probability value. It can be seen that *Project A* has a probability of less than 0.2 of having an effort overhead equal to or exceeding that of the "typical" and "majority" thresholds. Therefore, it is considered as being of low risk. On the other hand, *Project B* and *Project C* have a probability greater than 0.2 of exceeding the effort overhead for the typical projects and the majority of projects. Therefore, they are considered to be of high risk.

Note that the value of the acceptable risk level should again be determined by the most experienced project managers in conjunction with the quality assurance staff. It will be revised as more experience with the use of the model for risk assessment is gained. It should be remembered that acceptable risk is a business decision and should reflect the objectives and strategies of the organization as a whole.

#### Risk Assessment Based on Acceptable Risk Exposure

In the example presented in Fig. 7.7, the acceptable risk probability value was fixed as a common value across all risk levels and independent of potential "loss" in terms of effort overhead, meaning additional effort that needs to be spent in the project. This would mean that we are willing to accept higher exposure[4] to risk for the project with larger effort overhead. In order to maintain acceptable exposure to risk at the same level, we would define the acceptable risk probability value for high-risk projects (with large effort overhead) to be lower than the acceptable risk probability value for low-risk projects (with small effort overhead).

To address this issue, we will define acceptable risk in terms of *Acceptable Risk Exposure* (*ARE*) as opposed to a simple acceptable risk probability in terms of a fixed likelihood of exceeding a certain effort overhead value. Acceptable risk exposure is defined (7.2) as the product of acceptable risk probability (ARP) and effort overhead threshold ($EO_T$)

$$ARE = ARP \cdot EO_T \tag{7.2}$$

Please note that in order to explicitly distinguish between simple risk probability and the product of risk probability and potential loss, we introduce the term "risk exposure." Yet, in project risk management terminology, risk as such is defined as the product of an event's probability and potential loss (in contrast to simple event probability).

[4] We define risk exposure as the product of the probability of an undesired event and the potential loss if this event occurs. In our case, the undesired event is a project exceeding a certain effort overhead, and potential loss is the effort overhead.

**Risk Versus Risk Probability Versus Risk Exposure.**
In risk management, there are many different definitions of risk, which often leads to confusion. In the context of software project management, risk is defined as the product of an undesired event and the potential loss if the event occurs:

*Risk (undesired event)* = *Probability (event occurring)* × *Expected loss (after event)*

In order to explicitly differentiate between the simple probability of an undesired event and risk, we introduced two terms: risk probability and risk exposure. However, risk exposure actually corresponds to what is commonly referred to as risk.

In the ARE formula (7.2), acceptable risk probability (ARP) represents the probability of an undesired event and the effort overhead threshold ($EO_T$) represents the potential loss.

Let us illustrate the idea of risk exposure using the example we have been considering to explain previous risk assessment approaches earlier in this section. In Fig. 7.8, the acceptable risk exposure level for the threshold $T_1 = 45\ \%$ and the acceptable risk probability level 0.2 is equal to $0.2 \times 45\ \%$, which is approximately $ARE_1 = 9\ \%$. Acceptable risk exposure for the threshold $T_2 = 57\ \%$ and the 0.2 probability is equal to $ARE_2 = 11.4\ \%$. As we can see, the maximum acceptable risk exposure level is not constant across levels. Counterintuitively, for obviously high-risk projects (with large effort overhead) we are allowing greater risk exposure than for moderate-risk projects (moderate effort overhead) at the moderate-risk level before triggering risk management actions.

We may address this issue by setting acceptable risk exposure to be maintained constant across all risk classes determined by the effort overhead threshold (in our case three risk classes determined by the thresholds $T_1$ and $T_2$). Next, we would use a constant risk exposure level for recomputing the initially set acceptable risk probability level. Let us assume that we want to set risk exposure to the level of moderate-risk projects, $ARE_1 = 9\ \%$. The acceptable risk level for high-risk projects should then be set to $ARP_1 = ARE_1/T_2 = 0.16$ instead of 0.2, as it remains for the class of moderate-risk projects determined by threshold $T_1$. After modifying the acceptable risk probability level for $T_2$, the risk exposure for both $T_1$ and $T_2$ will be the same and equal to $ARE_1 = ARE_2 = 9\ \%$.

To determine the risk level using this approach, we follow the effort overhead curve just as in the approach based on acceptable risk level. We check in which risk interval (class) the curve crosses the acceptable risk probability or exposure level. Since these levels may differ across risk intervals, it is possible that the effort overhead curve intersects the risk probability or exposure levels in two or more different intervals. In such a case, we should classify the project into the highest risk class.

Figure 7.8 presents the analysis of a project's risk based on risk exposure for the three example projects: *A*, *B*, and *C*. If we consider the threshold-specific acceptable risk exposure levels $ARE_1$ and $ARE_2$, *Project A* will be classified as moderate risk,

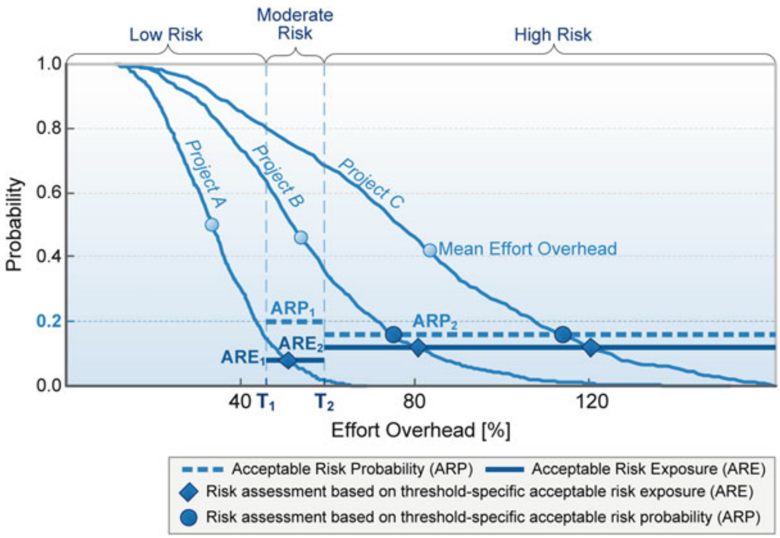


**Fig. 7.8** Example of risk assessment based on acceptable risk exposure level

whereas *Project B* and *Project C* will be classified as high risk. Now, if we take $ARE_1$ as the baseline acceptable risk exposure for all project risk classes, the acceptable risk probability level will be equal to $ARP_2 = 0.16$ for high-risk projects and remain at the initial level $ARP_1 = 0.2$ for moderate-risk projects. With respect to the threshold-specific acceptable risk levels $ARP_1$ and $ARP_2$, *Project A* would remain a low-risk project, whereas *Project B* and *Project C* would remain high-risk projects.

Note that the value of the acceptable risk level should again be determined by the most experienced project managers in conjunction with the quality assurance staff. This value will be revised to reflect experiences gained as well as changes in the business objectives and environment of the software organization applying the CoBRA method.

### 7.2.3 Risk Reduction

Once we have determined the risk level by applying one of the methods presented above, we may wish to identify those factors that have the strongest association with effort overhead in order to reduce risk. This information can be obtained through a so-called sensitivity analysis. This is an analysis of the actual contribution of the considered effort drivers to the effort overhead of a specific project. Figure 7.9 shows an example output of a sensitivity analysis. The contribution of the five most relevant effort drivers to the effort overhead for a specific project can then be used to drive risk reduction activities. For each of these factors, specific preventive and

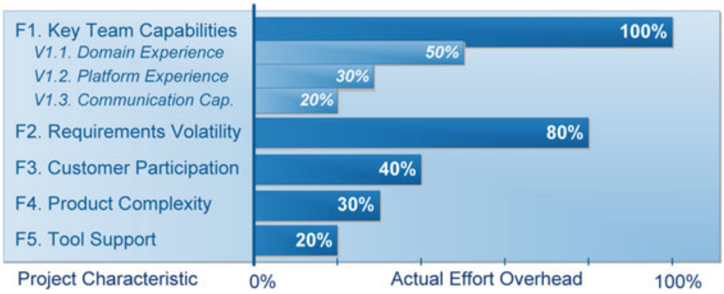


**Fig. 7.9** Example output of a CoBRA sensitivity analysis

corrective actions can be suggested in order to reduce risk. For this particular project, the "*Key Team Capabilities*" and "*Requirements Volatility*" factors have the strongest impact on project effort and would be the first targets of risk reduction activities. We may, for example, pay extra attention during project preparation to the capabilities of the project team and ensure that at least the key team members have necessary expertise and experience. In particular, we should consider the aspect of "*Domain Experience*," as the sensitivity analysis indicated the corresponding effort variable as being responsible for the greatest portion of effort overhead associated with the "*Key Team Capabilities*" effort factor.

Summarizing, we can say that in order to systematize the process of risk assessment and reduction outlined above, a set of guidelines for managing risks should be developed. These guidelines should consist of actions intended to reduce the impact of the identified effort drivers on each risk level. They should include typical types of responses to project risk (PMI 2007, Ch. 11) such as avoiding, transferring, mitigating, and accepting risk. Figure 7.10 presents the general steps of a simple effort-driven risk management approach.

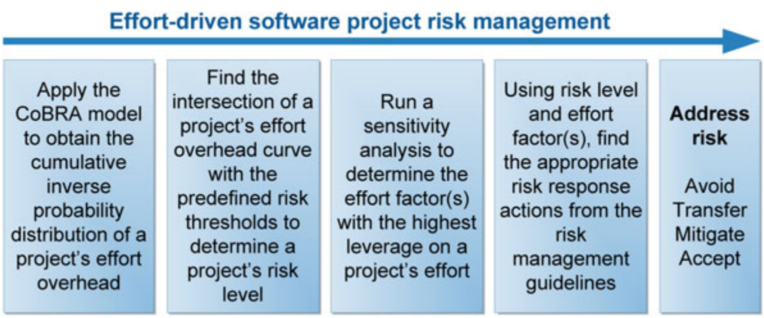


**Fig. 7.10** Simple risk management process

### Avoid Risk

*Avoiding risk* refers to actions the project manager should attempt to perform in order to entirely eliminate the negative impact of a certain effort factor on project effort.

### Mitigate Risk

*Mitigating risk* refers to actions the project manager should attempt to perform in order to reduce the probability and/or magnitude of the negative impact of a certain effort factor on project effort.

Avoidance and mitigation strategies use similar actions for addressing risk. In the context of effort estimation, risk avoidance would focus more on actions before project starts (thus offering a greater chance of entirely eliminating risk), whereas risk mitigation would focus on actions during the project. Typical actions include adjustments of the organization's processes or context characteristics. Globally, preventive actions are typically based on the impact of a certain effort factor over multiple projects. Recurrence of a certain factor with a negative impact on development effort may call for preventive actions to avoid it in the future. In this case, preventive actions focus on improving processes and environmental characteristics on the organizational level (organization-wide). Global changes make sense only when a specific process or context characteristic has been observed to consistently have a negative impact on project performance over all projects, independent of other project aspects. Locally, preventive actions are based on the expected negative impact of certain effort factors in the context of a specific project. Local actions typically make sense when a certain effort factor has a negative impact on project effort only in specific project conditions, for example, in conjunction with certain values of other effort factors.

**Example 7.1 Mitigating Project Risk Through Local Preventive Actions.**

Let us consider an example in which the project manager mitigates the risk of exceeding the acceptable project budget. The project manager mitigates this risk by improving those project characteristics the CoBRA estimation indicated as having the greatest negative impact on project effort.

The task of the project manager is to plan a new project so that it can be successfully completed within an *acceptable effort budget of 1,000 person-days*. The *acceptable risk level* was set at *Probability* = *0.2* of exceeding the acceptable effort limit. The CoBRA effort estimation model used in the project considers five effort factors, of which two are indirect factors and one is a composite factor. Figure 7.11 shows the effort overhead model of the CoBRA model used for estimating a new software project.

At the start of the project, the project manager assesses the values of the effort factors using their definitions and quantifications specified in the CoBRA model. Table 7.1 summarizes the definitions of the effort factors and their measured (assessed) levels for the new project.

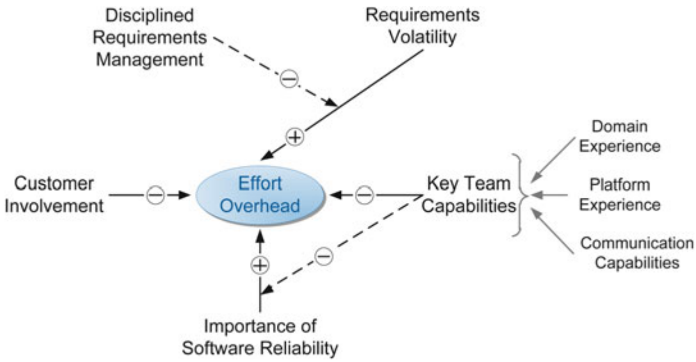


**Fig. 7.11** Example: effort overhead model

**Table 7.1** Example: effort factor data for estimated project

| Effort factor | Definition | Value |
|---|---|---|
| Key team capabilities | The extent to which the software development team possesses the skills and experiences necessary for the successful and efficient completion of the project (i.e., delivering software products of required functionality and quality within specified budget and time). | – |
| ***Domain experience*** | The extent of the project team's familiarity and comprehension of the target domain in which the developed software system is to be applied. | ***3*** |
| *Platform experience* | The extent of the project team's familiarity and comprehension of the platform for which the developed software system is intended. | 1 |
| *Communication capabilities* | The ability of the project team to communicate easily and clearly within the team (with other team members). | 1 |
| Requirements volatility | The extent to which the requirements are expected to change over time, after the requirements freeze. | 2 |
| Disciplined requirements management | The extent to which requirements are explicitly defined, tracked, and traced. This also includes the extent to which changes to requirements after their freeze are systematically managed (e.g., supported by the use of change management methods and tools). | 0 |
| Customer involvement | The extent to which the user/customer is involved in the project, providing necessary/useful information, reviewing requirements documents, performing some of the analyses themselves, and taking part in acceptance testing. | 1 |
| Importance of software reliability | The amount of attention that needs to be given to minimizing failures and ensuring that any failures will not result in safety, economic, security, and/or environmental damage, achieved through actions such as formal validation and testing, fault tolerant design, and formal specifications. | 1 |

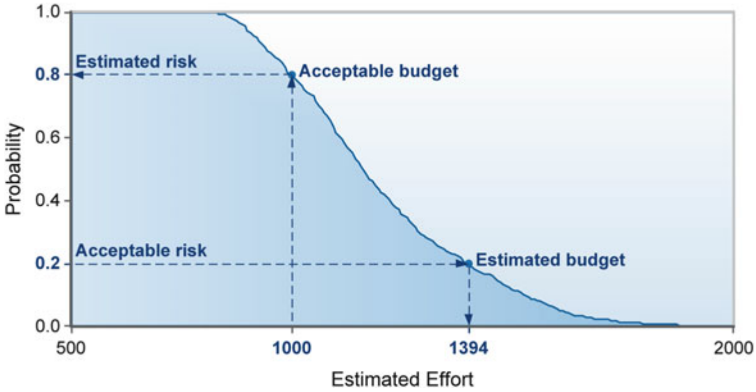


**Fig. 7.12** Example: distribution of initial effort estimate

Next, the project manager applies CoBRA model upon the data to obtain project estimates. Figure 7.12 presents the distribution of the initially estimated project effort. An analysis of the distribution indicates high risk (*Probability = 0.80*) of exceeding the acceptable project budget.

Mitigating the risk would require that the project manager either increases the project budget or the project's performance by improving those project characteristics that contribute to increased project effort. In order to stay within the acceptable probability (0.2) of exceeding the project budget, the budget would have to be increased to 1,394 person-days, which is almost 40 % more than the acceptable 1,000 person-days. Since increasing the project budget is not acceptable, the project manager has to mitigate the risk by increasing the project's performance.

In order to increase project performance and decrease its effort, the project manager looks at those project characteristics that contribute the most to increased project effort. For this purpose, the project manager runs a sensitivity analysis upon the CoBRA estimates and checks which of the effort factors considered in CoBRA model contribute the most to the project's effort overhead.

Since increasing the project budget is not possible, the project manager needs to identify the most promising improvement potentials with respect to the factors contributing to increased project effort. For this purpose, the project manager runs a sensitivity analysis upon the project data in order to check which effort factors considered in the CoBRA model actually have the greatest negative impact on the project effort. Figure 7.13 illustrates the results of the sensitivity analysis.

The results of the sensitivity analysis clearly indicate the dominant role of the project team's capabilities for successful project performance. From among the

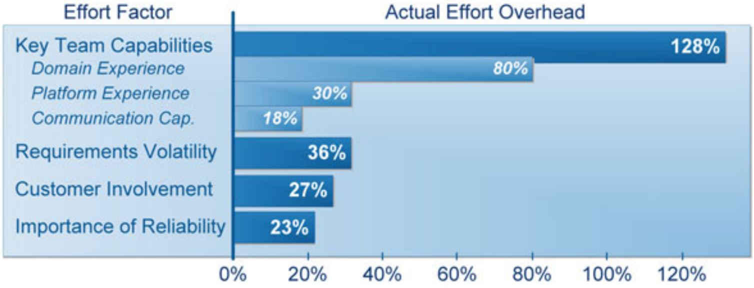


**Fig. 7.13** Example: sensitivity analysis

considered detailed capabilities, domain experience and platform experience have the greatest impact on project effort. These factors are first candidates for improvement; not only because they have the greatest impact on effort but also because the next most influential factors ("*Requirements Volatility*" and "*Customer Involvement*") depend on the customer and thus are rather difficult to improve.

The project manager decides to address this high project risk by first improving the domain experience of the key members of the project team. He achieves this objective by involving the domain experts in the key positions in the project and by providing domain training to the remaining team members. With the help of these means, the domain experience of the team improves dramatically from the worst level (factor value = 3) to the best level (factor value = 0). After improving the team's domain experience, the risk of exceeding the project budget deceases to $Probability = 0.42$. Still, this is more than acceptable level of 0.2. Figure 7.14 illustrates this improvement.

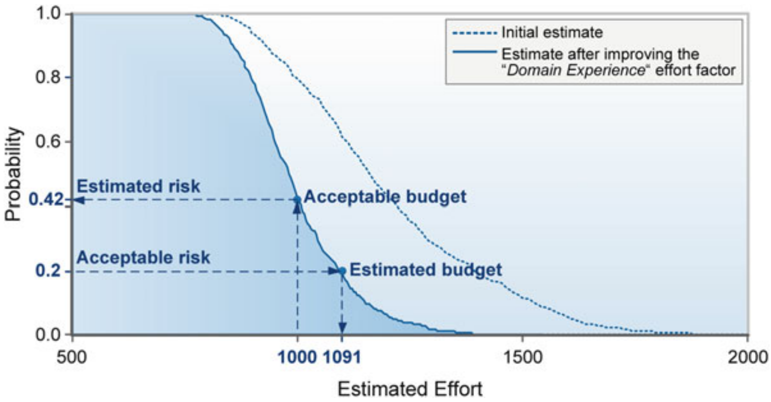


**Fig. 7.14** Example: estimated effort after improving domain experience

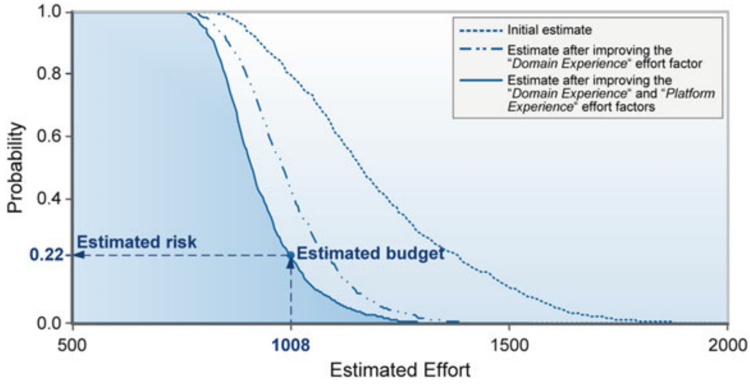


**Fig. 7.15** Example: estimated effort after improving domain and platform experience

If the project manager wants to keep the acceptable risk threshold, the project budget would need to be increased to 1,091 units, which is about 10 % over the acceptable project budget of 1,000 person-days. This is still an unacceptable solution. Therefore, the project manager decides to increase the project's performance by improving another project aspect represented by an effort factor in the CoBRA model. The project manager notices that the domain experts who are involved in the project to improve the domain experience of the team also have high platform experience. The only thing needed to increase the entire team's platform experience to the best level is appropriate training for the remaining team members. The project manager decides to include training in the project preparation phases. This way the level of the "*Platform Experience*" effort factor improves from 1 to 0.

As a result of improving both the "*Domain Experience*" and "*Platform Experience*" effort factors in the project, the risk of exceeding the acceptable project budget of 100 person-hours drops to $Probability = 0.22$ (Fig. 7.15). The actual risk level is just a little higher than the acceptable risk threshold of 0.2. In practice, the project could already be accepted. Although the project manager wants to meet the acceptable risk threshold by increasing the project budget, this would require increasing the budget by less than 1 % to 1,008 person-days. In practice, such an increase would probably also be acceptable.

Summarizing, it can be seen that the CoBRA method supports the project manager not only in identifying potential project risk but also helps him to reduce this risk by identifying the most important sources of risk and, related to that, the most promising means of risk mitigation. In practice, besides looking at the results of the sensitivity analysis, a project manager can simply play with the values of the effort factors he thinks he may improve and look at the outcomes of the estimation. In this trial and error way, the project manager can come up with a set of effort factor values that are necessary to meet the project's risk requirements. The discrepancy between expected and necessary factor values will then serve as a basis for the project planning and preparation activities. In our example, the project manager must staff the key project positions with experts in the application domain and in the platform and provide appropriate training to the remaining members of the project team prior to the start of the project. ■

Finally, a trivial, but often necessary, strategy to avoid or at least mitigate risk is to increase the project budget by a so-called contingency reserve to account for all negative impacts of relevant effort factors. In an extreme case, if the estimated project effort is not acceptable and the effort factors cannot be affected to decrease the effort, the project can be canceled, either before or after its start (preferably before).

### Transfer Risk

*Transferring* risk refers to actions that aim at shifting some or all of the negative impact of a certain effort factor outside the project or the organization. In the context of CoBRA effort estimation, one possible action would be to shift responsibility for improving customer-specific factors to the customer. For example, if customer involvement in the project is a critical effort factor, the customer should bear the consequences of his insufficient involvement in terms of increased project effort. Another possibility of transferring risk is to outsource risky project activities to a third-party organization. For example, if quality of testing is a critical effort factor and if the organization does not have sufficient expertise in testing, the testing activity can be entrusted to an independent company (this approach is known as independent verification and validation, IV&V). In this case, the IV&V organization takes over the risk of the testing activity, including the risk of keeping within the testing budget.

### Accept Risk

*Accepting risk* refers to a situation where none of the aforementioned three strategies can be used and accepting the risk "as is" is the only possibility left.

## 7.3 Project Scope Negotiation

Experiences we gained in industrial contexts indicate that customer involvement in software development is one of the factors that contribute significantly to overall development effort. Yet since software organizations typically have limited ability to affect this aspect, it is quite difficult to reduce the impact of this effort factor on project effort through internal improvement activities only. In practice, software project managers often face the situation where much of the project success depends on factors that are largely dependent on external parties. In this case, traditional risk mitigation and process improvement activities might not be effective because we have limited ability to influence the characteristics of an external entity involved in the project. At that point, we may consider two ways to prevent a project running into troubles:

- *Improvement of internal processes* (we discuss this aspect in Sect. 7.5). In this approach, we look for internal processes that may moderate the negative impact of the external party's characteristics on the project. In an effort model, this would be represented by interacting factors. Negative impact of insufficient customer involvement in the project may potentially be made less severe by improving the communication capabilities of the development team.
- *Negotiate project scope*. In this approach, we focus on those effort factors that refer to characteristics of external parties such as customers or external product/service providers. If one or more of these effort factors happen to be the source of large effort overhead, we can use this fact as an argument while negotiating the project conditions. For example, a software development company may require customer involvement in the project if finishing the project within the effort fixed by the customer largely depends on such involvement.

## 7.4 Project Benchmarking

The objective of project benchmarking in CoBRA is to compare software projects with respect to effort-related risks. In essence, in order to benchmark projects, we may use one of the risk analysis methods we presented for analyzing effort-driven project risks (Sect. 7.2). As a baseline for performing the benchmark, we take the risk thresholds we defined as percentiles upon the mean effort overhead of already completed successful projects. After setting up the thresholds, we can take one of the following benchmarking approaches:

1. Based on effort overhead thresholds: Comparing risk levels with respect to the mean effort overheads of benchmarked projects (Fig. 7.5).
2. Based on acceptable risk probability: Comparing risk levels with respect to the acceptable risk level assigned by an expert, for example, a quality engineer or project manager who is experienced in risk management (Fig. 7.7).
3. Based on acceptable risk exposure: Comparing risk levels with respect to risk exposure levels assigned by an expert (Fig. 7.8).

Figure 7.16 presents an example that illustrates the differences in the aforementioned three benchmarking strategies.

As we can see, depending on the risk assessment approach, projects may be assigned to different risk classes. Table 7.2 summarizes the classification of the three example projects with respect to the different risk assessment approaches.

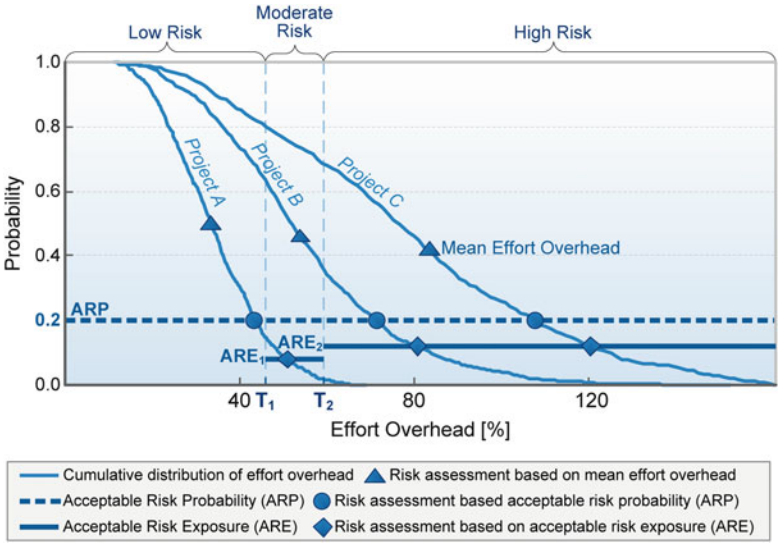


**Fig. 7.16** Comparison of various benchmarking strategies

**Table 7.2** Example benchmark with respect to project risk

| Risk assessment approach | Risk class | | |
|---|---|---|---|
| | Project A | Project B | Project C |
| Mean effort overhead | Low | Moderate | High |
| Acceptable risk probability | Low | High | High |
| Acceptable risk exposure | Moderate | High | High |

## 7.5 Process and Productivity Improvement

In principle, the goal of CoBRA modeling is to identify the most relevant effort dependencies. In other words, we look for project characteristics and their interactions that have the greatest impact on software development productivity[5] and effort. Running a sensitivity analysis on the effort model quantified for a specific project allows identifying those effort factors that actually have the greatest impact on the productivity and effort of this very specific project. In the short-term perspective, this information can be used locally, within the project, to avoid or mitigate project risks. When collected over multiple projects, this information can, in the long-term perspective, be used to drive process improvement activities. In this approach, we first identify processes that are indicated by effort factors that

[5] In practice, CoBRA can also be applied to model the effort of service-oriented software projects. In contrast to product development (product-oriented) projects, we would then refer to service efficiency instead of development productivity.

contributed the most to effort overhead across multiple projects. Next, we improve these processes in order to avoid large effort overheads in future projects.

**Example 7.2 Effort-Driven Software Process Improvement.**

Let us consider the example CoBRA effort overhead model in Fig. 7.17 and synthesized the results of the sensitivity analysis over the multiple historical projects in Fig. 7.18. We can see, for example, that the "*Key Team Capabilities*" make a consistent, significant contribution to the project costs. The sensitivity analysis indicates that, on average, 125 % of the project overhead is spent on overcoming the insufficient capabilities of key members of the project team. In order to decrease this additional effort and improve development productivity, an improvement of the organization's processes related to team capabilities is required.

In order to focus improvement actions, a detailed analysis of which team member and which capabilities exactly contribute most is required. In our example, definition of the factor provides first indication of the improvement area. Three specific key capabilities are considered here, of which the "*Domain Experience*" and "*Platform Experience*" factors have the greatest impact on effort. Next, the roles and activities in which these two capabilities are affecting

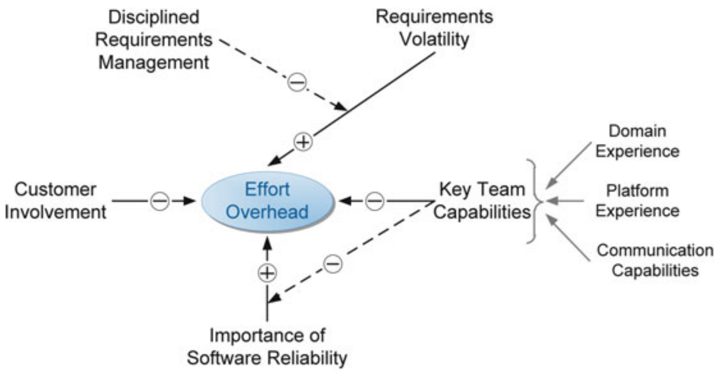


**Fig. 7.17** Example CoBRA effort causal model

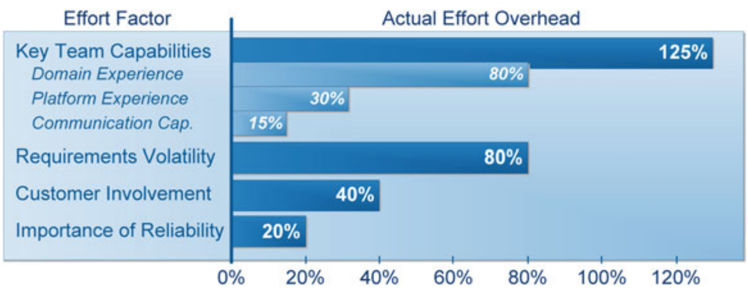


**Fig. 7.18** Example results of a CoBRA sensitivity analysis

development productivity the most should be identified, and related processes should be analyzed for possible improvements.

The second most influential factor is "*Requirements Volatility*." Here, the project manager may look for processes that are a potential source of volatile requirements and undertake appropriate improvement steps. Knowing, for example, that requirements specification is performed in a chaotic manner, the process group may decide about introducing systematic requirements specification processes. At the project level, the project manager can pay extra attention to this process and request extra provisions in the contract that the client commit to this process.

The project manager may, however, have little or no direct control over the sources of volatile requirements to reduce related effort overhead. In such a case, the effort overhead model suggests another solution. The manager may focus on improving "*Disciplined Requirements Management*," which alleviates the negative impact of volatile requirements on project effort. In principle, even though the effort model does not explicitly identify any useful factor interactions, the project manager can still identify indirect processes that moderate the negative impact of direct factors on project effort. ■

## Further Reading

- *A Guide to the Project Management Body of Knowledge—PMBOK Guide*, 4th Edition. Project Management Institute, Inc., 2008.

  PMBOK presents synthesized best-practice knowledge regarding project management. In particular, Chap. 11 of PMBOK summarizes basic approaches for managing project risks. The presented approaches may be used as a starting point for creating guidelines for managing effort-driven risks identified using the CoBRA method.

Springer

http://www.springer.com/978-3-642-30763-8

Software Cost Estimation, Benchmarking, and Risk Assessment
The Software Decision-Makers' Guide to Predictable Software Development
Trendowicz, A.
2013, XXVI, 322 p., Hardcover
ISBN: 978-3-642-30763-8