Organizational Risk Management –

The Balanced and Unbalanced Portfolio

Moshe Weiler, Ph.D.
Technion – Israel Institute of Technology
m_weiler@zahav.net.il

Yaron Arbetman
IBM R&D Labs in Israel
arbetman@il.ibm.com

Moshe Weiler: 5 Laskov St, Apt. #13, Netanya, 42656, Israel

Abstract

Many organizations encounter problems in constructing the organizational risk management from the projects' risks information in a bottom-up approach. The main problem dealing with the bottom-up approach, i.e., to collect the risks values from the projects, and accumulate them to the organization level, is how to reflect both, very high-edge risky projects and "more of the same" projects as well as "Mega" projects and small ones, on the same scale. In this article, we have suggested a new methodology that will give the organization's management this unique view on one chart. It will also help defining the norms of that organization in respect to which projects are within the "normal behavior" (Balanced Projects), and which projects are outside the "normal behavior" (Unbalanced Projects).

1. Introduction

1.1 Organizational Risk Management

Effective Risk Management is becoming necessary in today's dynamic world. It is also an important PA (Process Area) in process improvement models like the CMMI. In many cases, organizations are managing their financial and procedural risks on the organization level, and in parallel conducting a separate project risk management for each project. Creating a stronger connection between these two processes, can give additional advantages to the organization, as the projects risks have a direct impact on the organization risk level. In this article, we present a bottom-up approach for organizational risk management that is based on the projects risk management information. Most of the new ideas were targeted at large engineering based organizations (which conduct processes for concept definition, full-scale development and production), although these methods can suite other type of organizations, with some modifications (though, it seems less suitable for research-only organizations).

When constructing the organizational risk management from the projects' risks information in a bottom-up approach, we first need to define how to measure the project overall risk, based on the projects individual risks.

Then, we need to define how to bring up each individual project risks to the organization risk level, how to handle it in this level, and how to implement the feedback to the project level.

The methodology for defining the project risk level and deriving the associated organizational risks, includes elements that divided into the herein categories: Technology Risks, Schedule Risks, Cost Risks, and Programmatic Risks (risks that are beyond the level of the specific project, affect the project, and if necessary, must be mitigated at the organizational level).

In addition, we have the factor of the time horizon of the projects' risks. For example, if we recognize a continuous staffing risk (risk that is related to the Programmatic Risks category), in significant percentage of the projects, it might indicate a staffing risk in the organizational level, which should be handled also in a higher level than the project level. Analyzing and handling the resulted risk in the organizational level, can also improve managing the risk in the project level, when completing the top-down loop (see also section 2.4). In Section 3 we elaborate on the Programmatic Risks category and on the feedback loop to the project level.

1.2 The Problem

Based on the processes that should be achieved, a major issue that needs to be resolved is how to capture and handle the variance between different sized and different complexity projects. This problem is common in large organizations that focus all together on small, medium and large projects (sometimes, even Mega-projects) and execute risk management program per project. If we calculate a risk as the multiplication of its *Probability* times the *Impact* of it (calculated in a monetary value fashion), than normally, the higher management would focus on the expensive risks (in an absolute value). Is this the best strategy? We will show a new way that can enable large organizations to handle risk management programs for large, medium and small projects, with an accurate focus.

1.3 Resources vs. Complexity Model

One of the methods to manipulate this problem to another domain is to relate the Risk Management models to the Resources vs. Complexity models. The model of Moody [3], positioned a wide range of organizations over many types of projects on the Resources vs. Complexity chart (see Figure 1). This analysis gives an overview of the riskier projects vs. the more balanced projects. Since, in most cases, we aim to evaluate many projects in one organization (see section 1.2), our model will focus on one large organization that runs many types of projects in parallel.

1.3.1 Resources (following [3]). The Resources metrics are shown here, since we use them as a basis for modification suggested in a later stage. The scores of the horizontal axis of the Basic Chart represent a composite (sum) score of the following categories:

(a) Costs to develop the product through the first production unit (0-15);
(b) Time from the beginning of the effort through the first production unit (0-10);
(c) Infrastructure required completing the design (0-10).

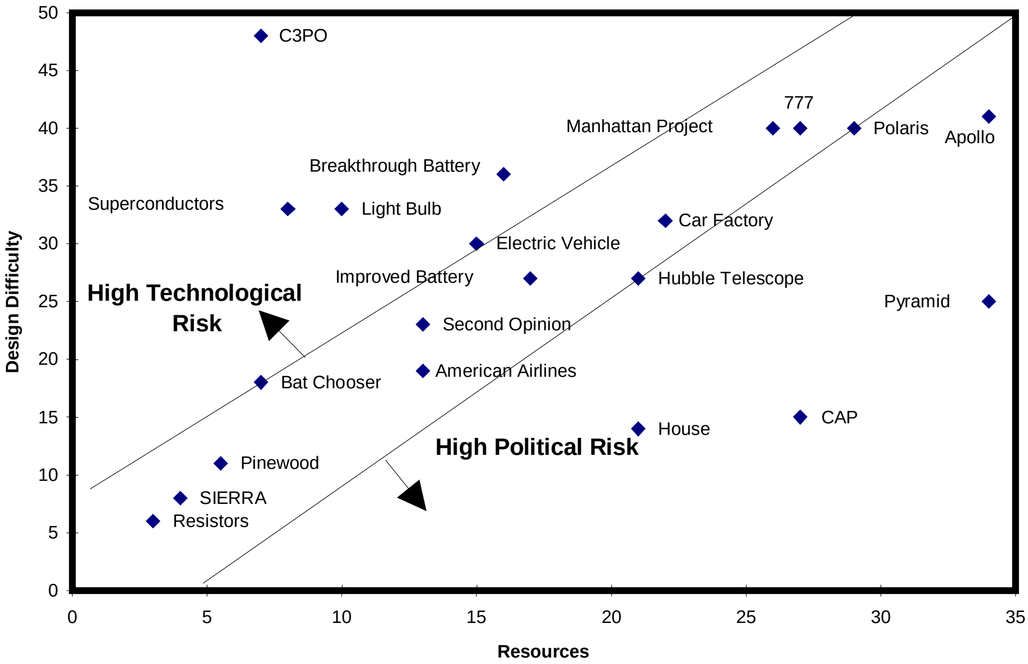

Figure 1. Risky projects in the Design Difficulty vs. Resources plane [3]

1.3.1.1 Cost Metric. Cost Metric measures the amount needed to pay for development, including salaries, utilities, suppliers, and materials, through the first production unit (see Table 1 for the Cost scores). The metric measures cost in terms of the payer's ability to pay and not in absolute dollars.

Table 1. Cost Metric scores

Points Range	Description
14-15	Massively expensive systems requiring major sacrifices
9-13	Very expensive systems that are rarely developed
3-8	Moderately expensive systems
0-2	Affordable systems

1.3.1.2 Time Metric. Time Metric measures the time spent from the beginning of the effort to define the customer's needs through the first production unit (see Table 2 for the Time scores).

Table 2. Time Metric scores

Points Range	Description
10	More than eight years
8-9	Five to eight years
4-7	One to five years
3	Six months to a year
2	Three months to six months
1	One to three months
0	Less than a month

1.3.1.3 Infrastructure Metric. Infrastructure Metric measures the physical resources needed for construction (including tools, process shops, and assembly workstations), transportation, communication, utilities, laws and legal protections, skilled managers, and the education and training system available (see Table 3 for the Infrastructure scores).

Table 3. Infrastructure Metric scores

Points Range	Description
9-10	Massive infrastructure requiring major portions of the available labor force and available equipment
6-8	Large complex infrastructure requiring large portions of the cost of the entire project
3-5	Moderate infrastructure requiring people on the project to support it
0-2	A common, low cost infrastructure

1.3.2 Complexity. Complexity is represented by the Design Difficulty scale [3], which includes six sub-metrics (not shown): Design type, Knowledge complexity, Steps, Quality, Process design, and Aggressive selling price. We will not use this scale in our model.

1.4 The Bonen Scale

Preliminary analysis of the maturity of a system in R&D projects was first introduced by Bonen [1], who classified design modules into four categories by the level of maturity they represented (see Table 4).

According to Bonen, at level 4, the project would need a separate research effort even before the project starts. The implications from the organization point of view are higher cost and more time spent, still not knowing if there is any viable solution. At level 3, there is a viable solution; still the project does not know how to reach there. At level 2, the project knows the solution; still a full R&D process is required. At level 1, small revisions are still required by the project. In our model, we used the *Modified Bonen Scale*, which will be described later.

Table 4. The Bonen Scale in R&D Projects

Level	Definition	Description
1	Revision or Variant Design	The project team is familiar with the solution (which has already been accomplished in-house), however small revisions are still required.
2	Engineering Gap or Adaptive Design	The project team knows what to do and is familiar with the solution; however, a full R&D effort is required.
3	Original Design or Viability Proof exists	The project team knows that a solution is feasible and that the technology exists, however the team does not know how to attain such a solution since it has never been attempted in-house before.
4	Research or No Viability Proof	The project team does not know whether or not a solution is possible or the technology available, research required.

2. Balanced and Unbalanced Project Portfolio

In order to represent different sized and complexity projects, a new methodology is suggested, based on calculating for each project not a single risk value (as the common practice), but two values on a two-dimensional chart: Resources vs. Complexity. For each of these factors we have suggested a calculating method. The new strategy combines two approaches from different aspects of systems engineering areas that have never been consolidated before. The measurement of required *Resources* is based on the modification of the model suggested by Moody [3], and the *Complexity* measurement is based on the *Modified Bonen Scale* (Hari and Weiss [2]).

2.1 Modifications of the Resources Metrics

The organization should calibrate and tailor its own Resources Scale (as well as the definitions, e.g., if a Government is the organization, an LCC cost shall be more suitable than the cost definition suggested here).

2.1.1 Modified Cost Metric. In our case (which is one organization and not many), the metric is normalized by default, i.e., the organization has its own data identifying what are considered as expensive systems and what are considered as affordable ones.

2.1.2 Modified Time Metric. In our case, the metric is normalized by the organization, based on its knowledge of what were the longest R&D effort and the common R&D effort (in terms of time-spent).

2.1.3 Modified Infrastructure Metric. In our case, the metric is normalized by the organization, based on the knowledge of what is a common, low cost infrastructure comparing to a large and complex infrastructure.

2.2 Modification of the Bonen Scale

As we mentioned earlier, the main disadvantage of Moody's model [3] is the complexity of the scale itself: while the *Resources* scale includes only three sub-metrics (Cost, Time, and Infrastructure that can be easily measured in the organization), the *Design Difficulty* scale includes six sub-metrics (Design type, Knowledge complexity, Steps, Quality, Process design, Aggressive selling price), which are very challenging to measure within the organization. For this axis, we suggested the use of the much clearer and easy to communicate *Modified Bonen Scale* [2]. This modified scale is divided into 5 categories (instead of the basic 4). The extra category is *Level 0*, which stands for "no extra design needed". In this level, the project team knows exactly what to do and what is the solution. It usually means that the production line of the system is stable, as compare to *Level 1* that might suffer from instability in the production line. The rest of the levels remain the same (see Table 5).

Table 5. The *Modified Bonen Scale* in R&D Projects

Level	Definition	Description
0	No extra design needed	The project team knows exactly what to do and what is the solution.
1	Revision or Variant Design	The project team is familiar with the solution (which has already been accomplished in-house), however small revisions are still required.
2	Engineering Gap or Adaptive Design	The project team knows what to do and is familiar with the solution; however, a full R&D effort is required.
3	Original Design or Viability Proof exists	The project team knows that a solution is feasible and that the technology exists, however the team does not know how to attain such a solution since it has never been attempted in-house before.
4	Research or No Viability Proof	The project team does not know whether or not a solution is possible or the technology available, research required.

2.3 The Balanced and Unbalanced Project Portfolio Model

The outcome of these joint models is a two-dimensional chart that enables us to identify projects that are inside or outside of the organizational *norms* (see in Figure 2 the two diagonal lines). There are three areas that a project may be positioned in the chart: (a) the area within the two diagonal lines that represents the norms of the organization, called the *Balanced Projects Area*; (b) Increased Political Risks, also called the Unbalanced Projects Area (projects with too many or wrong resources and too little complexity); (c) Increased Technological Risks, also called the Unbalanced Projects Area (projects with too much complexity and too little resources).

The main advantage of this representation is that it can handle all sorts of projects together, while the balanced projects are always in the main diagonal (from bottom left corner to the upper right corner as shown in the chart). The risky high technological projects are in the upper left corner (not enough resources to accomplish), while the high political projects are in the lower right corner (too many or wrong resources). It will also help defining the norms of that organization in respect to which projects are within the "normal behavior" (Balanced Projects) and which projects are outside the "normal behavior" (Unbalanced Projects). For example, Project E is balanced and "within norms", as well as Project D, although there is a big difference between the two. The first one is a common practice at the organization, and the second is one of a kind, yet, they are both balanced. On the other hand, Project B is in the High Technological Risk Area, and suffers from lack of resources, while Project G is in the High Political Risk Area and suffers from too many or wrong resources that are not needed. This method can also help in the CMMI levels 4-5, which requires a measurement of this *Process Area* (i.e., Risk Management) in the organizational level against some organizational norms.

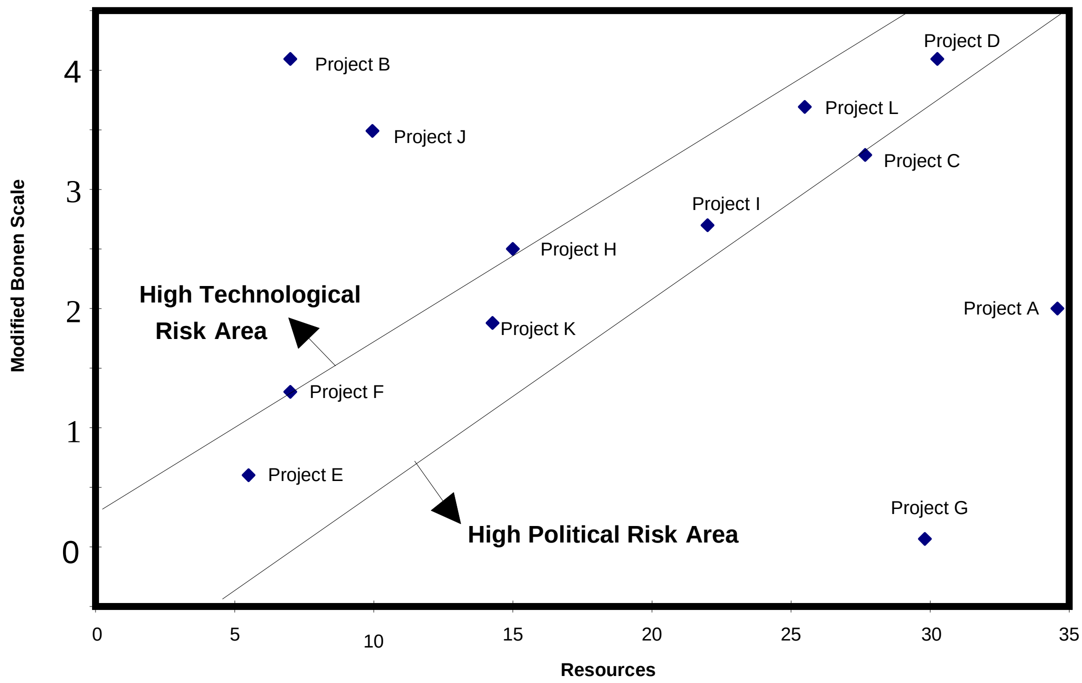

Figure 2. The Balanced and Unbalanced Project Portfolio Chart

2.4 Behavior over time

As higher management analyze the balanced and unbalanced projects chart over time, we might see two types of proper behaviors: (a) A project that is in the High Technological Risk Area should over time shift from upper left area in the chart to the main diagonals (i.e., management is adding necessary resources); (b) A project that is in the High Political Risk Area should over time shift from the right area in the chart to the main diagonals (i.e., management is cutting resources).

2.5 A Case Study

As part of this research, we have started a case study with IBM Laboratories in Israel. The results of applying the model (full or in part) within this case study will be presented in a future article.

3. Programmatic Risks Management

Until this point, we referred the organizational risk management as subject mainly to the resources and design complexity. In addition, we will address the category of Programmatic Risks that was described before: risks that are beyond the level of the specific project, affect the project, and if necessary must be mitigated at the organizational level. This category includes risks like staffing, single supplier, export permits, strikes, obsolete components, Government regulations, etc. The suggested model allows, in addition to the distribution between balanced and unbalanced projects, an analysis of the Programmatic Risks for each of the three areas: the Balanced Projects area, the High Technological Risk Projects area, and the High Political Risk Projects area. For each of the areas the Programmatic Risks are being calculated. The data are displayed according to the Programmatic Risk type (i.e., programmers staffing, obsolete components, single supplier, etc.), and the normalized risk factor (the risk category was calculated per project according to the common practice in the organization, i.e., Probability * Monetary Impact – in this case in M$), as presented for example in Figure 3 below (Projects A, B, C, D are in the size of $ 100M, $ 50M, $ 10M, $ 4M, respectively):

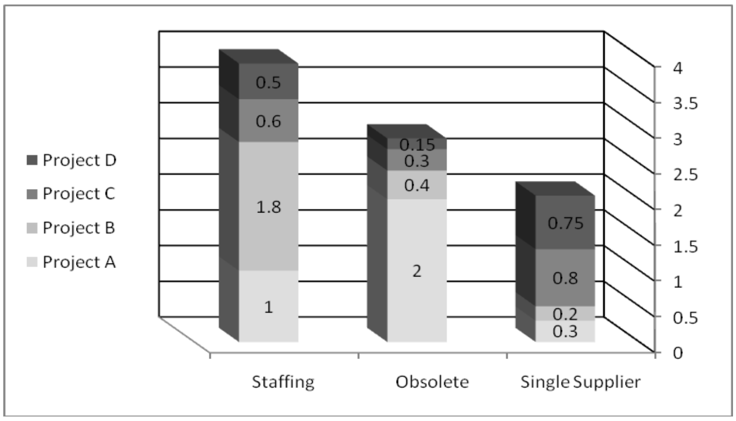

Figure 3: High Technological Risk Projects vs. Programmatic Risk Category

By looking at the Technological Risk Projects vs. Programmatic Risk Category chart higher management can see the herein results: (a) although Project A is a "Mega" project compare to Project D, it seems that in the *single supplier* category the monetary risk of the later is more than double compare to this risk in Project A. By using this approach the focus on Project D is immediate; (b) programmers staffing is the major Programmatic Risk (in terms of absolute monetary value). It seems that many programmers that do coding are missing in many projects; the organization can try and solve the root problem by outsourcing all of its coding tasks, or train enough programmers in-house to do the needed job, instead of trying to solve the problem project by project.

To gather this information, the project level information is being used. Having a central repository of project risks for all the projects in the organization enables automatic gathering of this information. For each project, as part of the project risk management, identification of the project risks according to risk type and risk severity is being maintained and managed. The risk severity is being calculated as multiplication of the risk probability, and the risk impact (in monetary value). Collecting the information from the projects level would enable to present the Programmatic Risks as suggested above. This method enables to focus more thoroughly on the Programmatic Risks from the organizational level, identifying organizational resources that are still needed in the High Technological Risk Projects, and exceeding organizational resources in the High Political Risk Projects (all per the level of knowledge that we hold at that point, as described in the Modified Bonen Scale).

The information can be analyzed in different organizational levels, according to the organization size, type, and number of projects. Analyzing of such information over time gives an additional benefit as it enables to observe and investigate the changes and trends in the programmatic risks histogram over time (as well as what mentioned in section 2.4).

4. Summary

In summary, such presentation of projects' risks in an organization is a beneficial and unique way to handle the complexity of the bottom-up approach. This suggested model is flexible enough to allow the definition of a risk management model suited to the organization environment, and at the same time has the advantage of improved identification and handling of projects risks in the organization level, and back down to the project level in a top-down approach. This strategy gives the organization a competitive edge at the current situation of many diverse risks in the world market.

The authors would like to thank *Bernard M. Gordon Center for Systems Engineering* for funding this research.

References

[1] Bonen, Z. 1964. “On the Planning of Development Projects”, Proceeding of the 3rd Conference on Operation Research.

[2] Hari, A., & Weiss, M. P. 2003. "Analysis of Risk and Time to Market During the Conceptual Design of New Systems", International Conference on Engineering Design ICED.

[3] Moody J. A. (et al.) 1997. *Metrics and Case Studies for Evaluating Engineering Designs*, Prentice Hall.

Biography

Moshe Weiler holds a Ph.D. in Systems Engineering from the University of Southern California (USC), Los Angeles (1998). He received his M.Sc. in Electrical Engineering from Tel Aviv University, Israel (1994), and his B.Sc. in Electrical Engineering from the Technion - Israel Institute of Technology, (1987).

Currently, he is a senior lecturer in the SE Program (for ME degree) at the Technion, a program that he was one of its founders (1999).

Dr. Weiler was appointed by INCOSE to be the first Ambassador to Israel (1997), and he was one of the founders of INCOSE_IL (the Israeli branch). He chairs the Risk Management Working Group (RMWG) at INCOSE_IL, and he is a researcher in the Gordon Center for SE at the Technion.

Yaron Arbetman holds a B.Sc. in Information Management from the Technion - Israel Institute of Technology, (1990). He has 15 years of experience in project management and software development in International hi-tech companies, and additional 5 years of experience in leading quality and process improvement activities.

Currently he is responsible for the quality and process improvement activities in the IBM R&D labs in Israel, and in the scope of this position he is also involved in promoting risk management activities. He is also a member of the Risk Management Working Group (RMWG) at INCOSE_IL.