Regular Paper

Risk Analysis of a Pinewood Derby: A Case Study

A. Terry Bahill[1],* and William J. Karnavas[2]

[1]*Systems and Industrial Engineering, University of Arizona, Tucson,* AZ 85721-0020

[2]IBM *Trans Arc Labs,* 11 *Stanwix Street, Pittsburgh,* PA 15222

Received March 2, 1999; revised October 1, 1999; accepted March 28, 2000

ABSTRACT

This paper presents risk analyses for Pinewood Derbies. It shows the derivation of probabilities and risk assessment numbers. It presents several risk analysis techniques and shows problems associated with them. © 2000 John Wiley & Sons, Inc. Syst Eng 3: 143–155, 2000

1. INTRODUCTION

Risk analysis is an important part of systems engineering. There are general guidelines for how it should be done, but there is no one correct way to do a risk analysis. This paper presents several risk analysis techniques and shows problems associated with them. By the end of the paper we think the reader will have seen enough examples of risk analyses to be able to do a risk analysis for a simple project.

This paper contains a simple case study that should be familiar. We think that most people will understand the derivation of our numbers and will relate to these every-day data. We hope that our readers can empathize with our problems and individual readers can actually implement some of our risk mitigation actions in their Cub Scout packs.

We ran a Pinewood Derby for our Cub Scout pack for 5 years. Each year we did a risk analysis, identified the most severe risks, and ameliorated them. For example, in the first year we used human judges. However, our judges had difficulty discriminating between cars crossing the finish line one-half inch apart. Therefore, in close races, they often declared *ties*. On the other hand, many parents thought that *they* were quite capable of distinguishing the true winner in close finishes: They always thought that *their* child should have been declared the winner. This produced unhappy parents. This was one of the first risks we eliminated. We switched to electronic judging, and this virtually eliminated irate parents. Every year we did a risk analysis and mitigated the biggest risks. In this paper we present the last of our risk analyses and a summary of the data from all 5 years. Of course, identifying *all* risks associated with any project is impossible. However, we show a wide variety of possible risks throughout this paper.

*Author to whom all correspondence should be addressed.

Systems Engineering, Vol. 3, No. 3, 2000

2. PINEWOOD DERBIES

Since the 1950s, over 100 million Cubs Scouts have built 5-ounce wooden cars and raced them in Pinewood Derbies. Pinewood Derbies have traditionally been single elimination tournaments where only the winner from each heat proceeded to the next round. This pleased scouts with fast cars, but for the unlucky majority it meant a single race, waiting for the awards to be announced, and then going home.

We changed the race format for our Cub Scout pack to a round robin, as shown in Table I, where each car is identified with a letter, e.g., A, B, C, ..., L. The objective was to allow each scout to race more often and race throughout the whole event. We decided to use six rounds, because that would give each car two races in each lane and still keep the whole event reasonably short. Switching from an elimination tournament to a round robin produced two side benefits: the scouts raced more of their friends and lane biases were ameliorated, because each car ran in each lane the same number of times.

We use the following terminology to describe Pinewood Derbies. Three cars running down the track at the same time is called a *heat*. The number of heats necessary for every car in the division (age group, e.g., Wolf, Bear) to run once constitutes a *round*. A set number of rounds (usually six) constitutes a *divisional contest*. Thus for 12 cars to run six times each, the divisional contest consists of six rounds of four heats each. Finally several divisional contests constitutes a *derby*.

Having decided to switch to a round robin, we now had to derive schedules. Table I shows a schedule for a 12-car, six-round, divisional contest. If there were only 10 cars in a divisional contest, then the 12-car schedule would be used, but no cars would be labeled K or L.

There are six mandatory requirements for the 12-car schedule: (1) Each car shall race in each of six rounds, (2) each car shall run twice in each lane, (3) there shall be three cars in each heat, (4) no cars shall race each other more than twice, (5) no car shall race without at least one opponent (even if cars K and L are missing), and (6) every car shall race every other car, except cars K and L shall not race each other. There are two preference requirements: (1) the first round should be in almost alphabetical order so that the scouts have some control over whom they race, and (2) minimize the number of pairs that race each other twice. The schedule in Table I satisfies these requirements.

Table I. A 12-Car Round Robin Schedule

	Lane 1	Lane 2	Lane 3
	Round 1		
Heat 1	A	B	C
Heat 2	D	E	F
Heat 3	G	H	K
Heat 4	I	J	L
	Round 2		
Heat 1	C	L	E
Heat 2	B	H	J
Heat 3	F	G	I
Heat 4	K	D	A
	Round 3		
Heat 1	K	I	C
Heat 2	G	E	B
Heat 3	J	F	A
Heat 4	H	L	D
	Round 4		
Heat 1	B	D	I
Heat 2	L	A	E
Heat 3	J	K	G
Heat 4	H	C	F
	Round 5		
Heat 1	C	J	D
Heat 2	F	B	K
Heat 3	E	I	H
Heat 4	A	G	L
	Round 6		
Heat 1	E	K	J
Heat 2	L	F	B
Heat 3	D	C	G
Heat 4	I	A	H

More schedules and discussions about schedules are given in Chapman, Bahill and Wymore [1992], Bahill and Karnavas [1993], and Moody, Chapman, Van Voorhees, and Bahill [1997]. The systems engineering of a Pinewood Derby is given in Chapter 5 of Chapman, Bahill, and Wymore [1992]. A revised version is available at http://www.sie.arizona.edu/sysengr/pinewood/pinewood.pdf. A preliminary failure modes analysis of this system is given in Section 7.1.2 of Chapman et al. [1992].

3. RISK ANALYSIS

Risk is often divided into four broad categories: performance, schedule, cost and safety. First, we will look at performance, which is often called technical performance.

3.1. Performance Failure Modes

We considered the following performance failure modes (also called hazards):

failure of sensors at the top or bottom of the track;
track imperfections that cause

one lane to be faster than another (called lane bias) and
a car to jump out of its lane and collide with another;
mistakes in finish line judging and recording of results; and
human mistakes in
weighing the cars,
allowing car modifications after inspection,
placing cars in the wrong lanes,
lowering the starting gate,
resetting the finish line switches, and
wasting time.

Table II presents a failure modes and effects analysis (FMEA) for the performance aspect of a Pinewood Derby using a Round Robin (as in Table I) with electronic judging. Each car's fastest time for its six races was used to determine first, second, and third places in each division. We assume one heat per minute and one car inspection per minute. So the *Probability of Failure* is the probability of failure per minute, or per heat, or per car. *Severity* (sometimes called Consequences) indicates how badly the event would be disrupted if this failure occurred; 1 means no effect and 10 means a big disaster. Intermediate levels for severity will be given later in this paper. In general, severity is a combination of the dreadfulness of the failure, how well the failure is understood, and the number of people it effects. *Difficulty of Detection* indicates the difficulty of devising and executing tests that will detect each failure; 1 means it is very easy and 10 means it is very difficult. The *Risk Priority Number* is the product of Probability of Failure, Severity, and Difficulty of Detection. Higher risk priority numbers imply a greater danger of shipping a defective product, or in this case running a defective Pinewood Derby.

3.1.1. Rationale for the Performance Failure Probabilities

Performance Failure Probabilities were computed on a failures per heat basis. We collected data during the five Pinewood Derbies that we ran. Each year had (in round numbers) 100 cars and 200 heats.

Temporary failure of sensors at top or bottom of track. This happened about twice a year, yielding a probability of 10^{-2} failures per heat.

Lane biases. We think all Pinewood Derby tracks have one lane faster than another. Thus our probability of lane bias is 1.

Collisions between cars. Historically, about one out of ten heats had collisions, yielding a probability of 10^{-1}

Table II. Performance Failure Modes and Effects Analysis for a Pinewood Derby

Failure Mode	Potential Effects	Probability (failures per heat)	Severity	Difficulty of Detection	Risk Priority Number
Temporary failure of sensors at top or bottom of track	Heat must be re-run	10^{-2}	1	1	10^{-2}
Lane biases	Some cars have unfair advantage	10^{-0}	0	9	0
Collisions between cars	Heat must be re-run	10^{-1}	1	1	10^{-1}
Mistakes in judging or recording	Wrong car is declared the winner	10^{-5}	6	10	6×10^{-4}
Human mistakes in					
weighing cars	Some cars have unfair advantage	3×10^{-2}	2	5	3×10^{-1}
allowing modifications	Some cars gain unfair advantage	1.5×10^{-2}	3	7	3×10^{-1}
placing cars in wrong lanes	Wrong car is declared the winner	2×10^{-2}	6	6	7×10^{-1}
lowering the starting gate	Cars can be broken, race could be unfair	10^{-2}	1	1	10^{-2}
resetting finish line switches	Heat must be re-run	10^{-2}	1	1	10^{-2}
wasting time	People get annoyed	10^{-2}	4	1	4×10^{-2}

failures per heat. Collisions were probably caused by imperfections where two sections of track were joined. When a wheel hit such an obstruction, the car bounced out of its lane. By carefully aligning and waxing the joints, we got the failure rate down to 1 out of 40 heats.

Mistakes in judging or recording results. Our estimate is 10^{-5}.

Human mistakes in weighing cars. We think that every year an overweight car snuck through. This car would then have run in six heats, meaning 6/200 heats had an overweight car, 3×10^{-2}.

Human mistakes in allowing modifications (such as adding graphite) *after inspection.* If each year one scout added graphite in the middle of his six heats, then our probability of an unfair heat would be 3/200, 1.5×10^{-2}.

Human mistakes in placing cars in the wrong lanes. We detected this four times in one derby, 2×10^{-2}.

Human mistakes in lowering the starting gate. This happened about twice per derby, 10^{-2}.

Human mistakes in resetting finish line switches. This happened about twice per derby, 10^{-2}.

Human mistakes producing wastage of time. This happened once or twice per derby, 10^{-2}.

3.1.2. Discussion

The most important failure modes are *human mistakes in (1) placing cars in the wrong lanes, (2) allowing modifications after inspection, and (3) misweighing cars.*

In a previous failure analysis of our Pinewood Derby using quality function deployment [Chapman, Bahill, and Wymore, 1992: Chapter 7], we found that in order to satisfy our customer we had to pay the most attention to *Mistakes in Judging or Recording* and *Lane Biases*. Therefore, we designed our system to pay special attention to these failure modes. We designed a computerized judging and recording system that had a very low probability of failure, indeed this failure mode is the least likely one in Table II. Furthermore, we designed a round robin tournament where each car races twice in each lane; therefore, lane biases had no effect. When the effect of a failure mode was eliminated, we set the severity to 0. We did not remove the failure mode, because we wanted to show that it was considered. Therefore, the failure modes that were very important in our early failure analysis were not important in this failure analysis. In our redesigned Derby, humans were the most important element.

Failure modes and effects analyses typically do not include possibilities of human failure. We have included them in our analysis, because we think that if the process is designed with these items in mind, then the system can better accommodate such failures. As long as fallible humans are involved, we will have mistakes. So now our problem is, can we design a system that will perform well in spite of human mistakes?

The original FMEA standard is MIL-P-1629 published in 1949[FMEA, 1949]. Since then a lot of work has been done on FMEA. The Automotive Industry Action Group (AIAG) and the Society of Automotive Engineers have developed standards, respectively AIAG FMEA and SAE J1739. There are web sites to help with FMEA (e.g., http://www.fmeca.com/ffmethod/history.htm) and web sites listing dozens of commercially available software systems that can help implement FMEA (e.g., http://www.enre.umd.edu/ffp.htm).

Other Columns. A failure modes and effects analysis should include a column indicating who should do what in response to each failure mode. We did not include such a column, because it would not have been useful in this analysis. Each entry would have simply said Bahill or Karnavas fixes it. To do a root-cause analysis, we could have included "So what?" columns. For example, the failure mode *Collisions between cars* has the potential effect of "Heat must be rerun." After this we could ask "So what?" which might produce the response, "This takes additional time." We could ask again, "So what?" producing, "If there were many reruns, the whole Pinewood Derby schedule might slip," etc. Other columns could also be used to document what and when corrective actions were taken.

Considering Difficulty of Detection was a good idea, but it is not common in the systems engineering community. So we will not consider it anymore. Therefore, from now on our Estimated Risk will be the Probability of Failure times the Severity.

Some people have used the Probability of Failure plus Severity minus the product of Probability of Failure and Severity. But this formula does not perform satisfactorily. For example, if you set the severity to 1 (assuming a range of 0–1), then the probability of failure could be reduced from say 10^{-1} to 10^{-6} without changing the risk. We do not want this. Therefore, we do not use this technique.

Table III shows our Performance Failure Modes and Effects Analysis without the Difficulty of Detection. The most important failure modes are *(1) human mistakes in placing cars in wrong lanes and (2) collisions between cars.* These are slightly different than in the previous table.

However, this failure modes analysis is flawed, because the probability of failure runs over six orders of magnitude from 10^0 to 10^{-5}, whereas severity only ranges over one order of magnitude from 1 to 10 (the 0 for Lane Biases is for a previously solved failure mode). This means that the probability of failure is the dominant discriminator in this failure modes analysis.

Table III. Second Performance Failure Modes and Effects Analysis

Failure Mode	Potential Effects	Probability (failures per heat)	Severity	Estimated Risk
Temporary failure of sensors at top or bottom of track	Heat must be re-run	10^{-2}	1	10^{-2}
Lane biases	Some cars have unfair advantage	10^{-0}	0	0
Collision between cars	Heat must be re-run	10^{-1}	1	10^{-1}
Mistakes in judging or recording	Wrong car is declared the winner	10^{-5}	6	6×10^{-5}
Human mistakes in				
weighing cars	Some cars have unfair advantage	3×10^{-2}	2	6×10^{-2}
allowing modifications	Some cars gain unfair advantage	1.5×10^{-2}	3	4×10^{-2}
placing cars in wrong lanes	Wrong car is declared the winner	2×10^{-2}	6	10^{-1}
lowering the starting gate	Cars can be broken, race could be unfair	10^{-2}	1	10^{-2}
resetting finish line switches	Heat must be re-run	10^{-2}	1	10^{-2}
wasting time	People get annoyed	10^{-2}	4	4×10^{-2}

Consider Table IV. The examples in the left and right halves have the same probability of failure, but the severity column in the right half has been turned upside down. The risk columns are different, but the rank order columns are identical. Severity had no effect! This shows that the figure of merit with the biggest range dominates the result.

Therefore, in Table V, we have expanded the severity so it also runs over six orders of magnitude. Thus probability of failure and severity have the same range, so that they have equal impact. The most important failure modes in this analysis are *human mistakes in (1) placing cars in the wrong lanes and (2) wasting time.*

Risk communication is an important part of a risk analysis. What do we communicate to whom? In general, we tell the program manager what the most significant risks are, and we offer proposals for mitigating these risks. In this case, we tell the Pinewood Derby Marshall that *placing cars in the wrong lanes* is the most significant risk. Then we point out that the race schedules (as in Table I) have lane 1 on the left and lane 3 on the right. This matches the perspective of the finish line judges (lane 1 is on their left and lane 3 is on their right), but it is the opposite for the starter at the top of the track. Therefore, we suggest printing regular schedules for finish line judges and printing mirror image schedules in another color for use by the starter.

Risk is often presented graphically with severity on the ordinate and probability of failure on the abscissa.

Table IV. The Problem with Different Ranges

Example 1				Example 2			
Probability	Severity	Risk	Rank Order	Probability	Severity	Risk	Rank Order
10^{-1}	1	1×10^{-1}	1	10^{-1}	6	6×10^{-1}	1
10^{-2}	2	2×10^{-2}	2	10^{-2}	5	5×10^{-2}	2
10^{-3}	3	3×10^{-3}	3	10^{-3}	4	4×10^{-3}	3
10^{-4}	4	4×10^{-4}	4	10^{-4}	3	3×10^{-4}	4
10^{-5}	5	5×10^{-5}	5	10^{-5}	2	2×10^{-5}	5
10^{-6}	6	6×10^{-6}	6	10^{-6}	1	1×10^{-6}	6

Table V. Third Performance Failure Modes and Effects Analysis

Failure Mode	Potential Effects	Probability (failures per heat)	Severity	Estimated Risk
Temporary failure of sensors at top or bottom of track	Heat must be re-run	10^{-2}	10^{1}	10^{-1}
Lane biases	Some cars have unfair advantage	10^{0}	10^{0}	10^{0}
Collision between cars.	Heat must be re-run	10^{-1}	10^{1}	10^{0}
Mistakes in judging or recording	Wrong car is declared the winner	10^{-5}	10^{6}	10^{1}
Human mistakes in				
weighing cars	Some cars have unfair advantage	3×10^{-2}	10^{2}	3×10^{0}
allowing modifications	Some cars gain unfair advantage	1.5×10^{-2}	10^{3}	1.5×10^{1}
placing cars in wrong lanes	Wrong car is declared the winner	2×10^{-2}	10^{6}	2×10^{4}
lowering the starting gate	Cars can be broken, race could be unfair	10^{-2}	10^{1}	10^{-1}
resetting finish line switches	Heat must be re-run	10^{-2}	10^{1}	10^{-1}
wasting time	People get annoyed	10^{-2}	10^{4}	10^{2}

Each of the axes is *linear* with a range of 0–1. Risks in the upper right corner are the most serious and should be handled first. The problem with this technique is that all of the risks in Table V would be squashed onto the *x*-axis, because they all have probabilities at or below 10^{-1} (except for our anomalous lane bias). If some action resulted in a reduction of probability of failure from 10^{-2} to 10^{-4}, we want the estimated risk to change. And in these graphs it would not change. Using logarithmic scales would help, and indeed this may be the best solution if quantitative data are available. However, we have not seen this used in the systems engineering literature.

What should we learn from this example? (1) The ranges for probability of failure and severity should be the same, unless different ranges are deliberately being used as a means of weighting. (2) Linear scales are not useful when most of the data are small numbers like 10^{-x}. We will show one way of handling these problems in the next section.

3.2. Schedule Risk

We are now ready to evaluate the Schedule Risk. We want the probability that all parts of the system function so that any given heat can be run.

We considered the following schedule failure modes:

total loss of electric power;
two types of computer hardware failure,
  personal computer failure during the race and
  the server goes down before schedules are printed;
computer software failure in
  commercial software,
  custom software, and
  interfaces;
adverse weather conditions; and
forgetting equipment.

The probabilities given in Table VI represent the probability that a particular failure will cause a delay of one heat.

3.2.1. Rationale for the Schedule Risk Probabilities

Schedule Risk Probabilities were computed on a failures per minute basis. Because we ran a heat per minute, this is the same as the probability of delaying an individual heat. Lots of other criteria could have been used, such as: Did the Derby start on time? Did it end on time?

Did each Divisional Contest start on time? Did each end on time?

Table VI. Schedule Risk Analysis for a Pinewood Derby

Failure Mode	Potential Effects	Probability (failures per minute)	Severity	Estimated Risk
Loss of electric power	Delay races until electricity is restored	2×10^{-5}	10^2	2×10^{-3}
Personal computer hardware failure	Cannot compute anything, cancel the Derby	10^{-3}	10^5	10^2
Server failure	We do not have schedules	10^{-8}	10^4	10^{-4}
Failure of commercial software	Cannot compute anything, cancel the Derby	10^{-5}	10^4	10^{-1}
Failure of custom software	Cannot compute winners	5×10^{-5}	10^3	5×10^{-2}
Software interface failure	Data is lost and some races must be re-run	2×10^{-5}	10^1	2×10^{-4}
Bad weather	NiCad batteries lose power, cannot determine winners	8×10^{-6}	10^3	8×10^{-3}
Forgetting equipment	We lose an hour of setup time	10^{-5}	10^3	10^{-2}

Loss of electric power. In the last 14 years, this area of Tucson has lost electric power three times (once when lightning hit a transformer, once when a car hit a pole, and once when the Western Power Grid went down), for a total outage of almost 3 h. Thus the probability that the power was out for any given minute was 2×10^{-5}.

Personal computer hardware failure. During the last 5 years, thrice Bahill has had hardware fail while he was using a personal computer. It took a total of 45 h to fix it or get a replacement, 10^{-3}. This does not count periods of upgrading hardware or software.

Server failure. We printed schedules weeks in advance, so server failure was not apt to delay a race, 10^{-8}.

Failure of commercial software. 105: This probability of failure is low, because our programs were Unix- and DOS-based.

Failure of custom software. 5×10^{-5}.

Software interface failure. In 5 years we had one interface failure: It took 1 h to fix, hence 2×10^{-5}.

Bad weather. In the last 14 years Tucson has had one afternoon where the temperature dropped 30° in 2 h. This caused the NiCad batteries to lose power, and we shifted to a manual, paper-based system. The switch took 1 h, yielding 8×10^{-6}.

Forgetting equipment. Over the last 14 years, on one road trip we forgot equipment and had to go back to the lab to get it. This caused a delay of a little over 1 h, 10^5.

The probability that this system would be operational for any given heat is $(1 - 2 \times 10^{-5})(1 - 10^{-3})(1 - 10^{-8})$ etc., which equals 0.99888. This is good reliability. It resulted from doing studies like these, finding the weak link, and redesigning the system. As you can see in this table the present weak link is *Personal computer hardware failure.* Therefore, when we put on a Pinewood Derby, we keep a spare computer in the car. Actually we are even more paranoid than this. The last time we put on a Pinewood Derby, we designed it with a Round Robin (Best Time) with electronic judging, but we also designed a backup system that required no electricity: It used a Round Robin (Point Assignment) with human judging. All of the equipment and forms for the backup system were in the car.

Failure modes and effects analyses usually do not include acts of nature, like our *Bad weather* category. We have included it in our analysis, because we think that if the process is designed with these items in mind, then the system can better accommodate such failures.

3.2.2. Qualitative Scales

From the Performance Failure Modes and Effects Analysis we learned that the ranges for probability of failure and severity should be the same (unless different ranges are deliberately being used as a means of weighting) and that mixing real probability numbers with subjective measures of severity is dangerous. One solution for this problem is making both the probability and the severity numbers between 1 and 10 with qualitative descriptions. Of course, it does not make a difference if

Table VII. Qualitative Scale for Probability of Failure

What is the Probability that the risk will happen?	Your processes...	Level
Not probable	...will effectively avoid or mitigate this risk based on standard practices.	1 or 2
Low probability	...have usually mitigated this type of risk with minimal oversight in similar cases.	3 or 4
Probable	...may mitigate this risk, but workarounds will probably be required.	5 or 6
Highly probable	...cannot mitigate this risk, but a different approach might.	7 to 9
Near certainty	...cannot mitigate this risk, no known processes or workarounds are available.	10

the range is [0–1], [1–5], [1–10], or [1–100]. Use whatever your customer is most comfortable with. What is important is that both have the same range. The qualitative scales given in Tables VII and VIII are based on Boeing [1997], McDonnell Douglas [1997], Moody et al. [1997], and Blanchard and Fabrycky [1998].

Using scoring functions [Chapman, Bahill, and Wymore, 1992; Wymore, 1993] would be a better, but more complex solution to the problem of unequal ranges.

Table VIII. Qualitative Scales for Severity

If the risk happens, what would be the Severity of the impact?			
Performance	Schedule	Cost	Level
Minimal or no impact	Minimal impact, slight changes compensated by available program slack	Cost targets not exceeded	1 or 2 (low)
Minor reduction in technical performance, same approach retained	Additional activities required, able to meet key dates	Cost estimates exceed target by 1 to 5%	3 or 4 (minor)
Moderate reduction in technical performance, but workarounds are available	Minor schedule slip, will miss minor milestone	Cost estimates exceed target by 5 to 20%	5 or 6 (moderate)
Significant degradation in technical performance, workarounds are difficult	Program critical path affected	Cost estimates exceed target by 20 to 50%	7 to 9 (significant)
Technical goals cannot be achieved	Cannot achieve key program milestone	Cost estimates exceed target by more than 50%	10 (high)

Table IX. Second Schedule Risk Analysis

Failure Mode	Potential Effects	Probability value	Severity	Risk Assessment
Loss of electric power	Delay races until electricity is restored	4	2	8
Personal computer hardware failure	Cannot compute anything, cancel the Derby	6	8	48
Server failure	We do not have schedules	1	6	6
Failure of commercial software	Cannot compute anything, cancel the Derby	4	6	24
Failure of custom software	Cannot compute winners	4	4	16
Software interface failure	Data is lost and some races must be re-run	4	1	4
Bad weather	NiCad batteries lose power, cannot determine winners,	2	4	8
Forgetting equipment	We lose an hour of setup time	4	4	16

We will now use the numbers in Tables VII and VIII to redo the Schedule Risk analysis (see Table IX). This analysis shows that the most important schedule risks are (1) personal computer hardware failure and (2) failure of commercial software, which, fortunately, is the same result that we obtained with the exponential numbers. Therefore, we are going to continue to use these qualitative scales.

Probabilities can be calculated quantitatively. If you have probabilities that run over many orders of magnitude, it is a shame to compress them into a range of 1–10. But, on the other hand, severity is almost always a qualitative assessment, and the 1–10 scale is natural. And we have seen that mixing probabilities that run over six orders of magnitude with severities that run from 1 to 10 gives undue weight to the probabilities. There are three simple solutions to this dilemma: Make both run over six orders of magnitude (using logarithmic scales for graphs), make both run from 1 to 10, or explicitly use weights of importance for both. It is up to the system engineer to choose one of these techniques and apply it.

3.3. Cost Risk Analysis

We analyzed the Cost Risk from the viewpoint of the people who were designing the Pinewood Derby in the month before the derby. Once, during design and construction, we discovered that our bottom-of-track sensors did not work satisfactorily. So we had to design and build a different sensor system, which cost $150 and required an additional 25 h. The sensors in service could also burn out or break, which would take $10 and 2 h to fix. Also our computers might break, in which case we would have to repair or replace them. Historically, repairing each computer field failure has cost $100 and 15 h (see Table X).

Table X. Cost Risk Analysis for a Pinewood Derby

Failure Mode	Potential Effects	Probability value	Severity	Risk Assessment
Sensor system fails to meet requirements	Design and build different sensor system	6	8	48
Sensor burns out or breaks	Replace sensor	4	6	24
Computer breaks	Fix computer	3	7	21

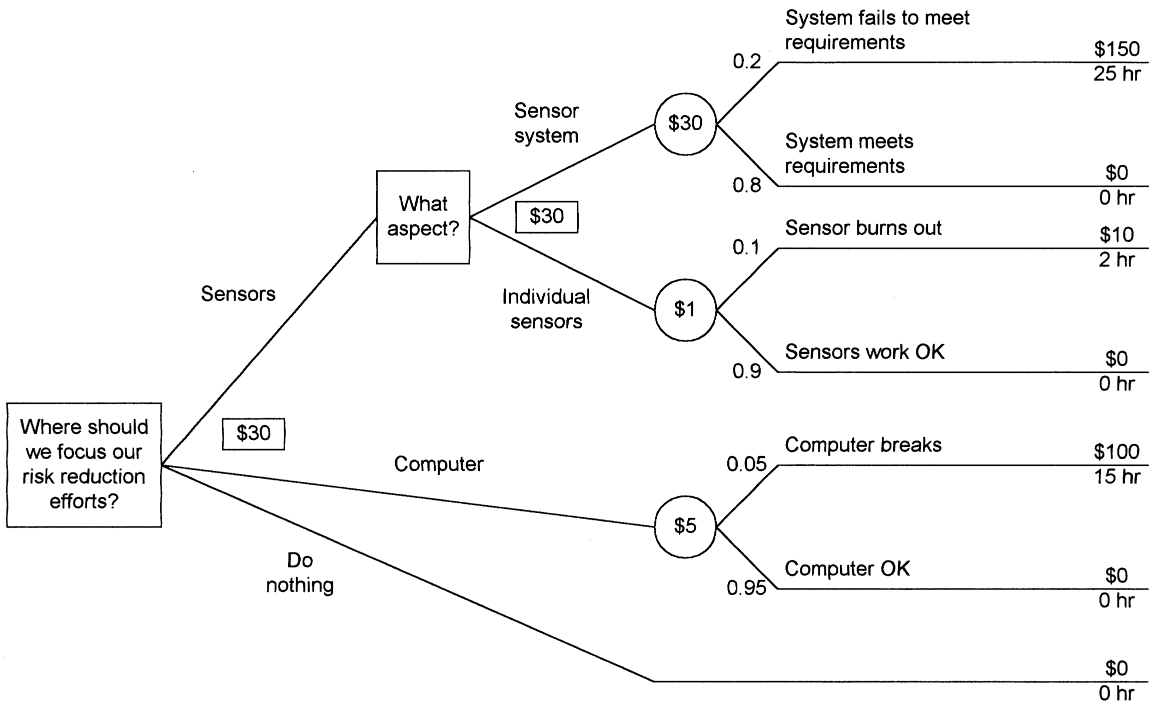

Figure 1. Decision tree for the cost-risk analysis.

Figure 1 shows a decision tree [Hall, 1999; Kirkwood, 1999] for this cost risk analysis. The decisions to be made are in boxes. The chance nodes are circles containing the expected value of the events to the right. Outcomes of the chance nodes are labeled with event probabilities. The expected values are derived from right to left. The largest expected value, $30, is for the sensor system failing to meet its requirements. This means that we would be best off devoting our risk reduction efforts to the design and implementation of the system of sensors at the bottom of the track that detects the cars crossing the finish line. In this figure we only computed the expected values using the dollar costs, but the hourly cost data yield the same conclusion.

3.4. Safety Risk

We also considered Safety Risk. We were concerned with injury to humans (both physical and psychological), pinewood derby cars, and equipment. Pinewood derby cars are fragile. If one falls off the top of the track, it will break, the scout will be out of the Derby, and the scout will feel very bad (see Table XI).

We should use this information to allocate resources. We should use money to buy a safety net for the cars, before buying shock proof boxes for the instruments.

Table XI. Safety Risk Analysis for a Pinewood Derby

Failure Mode	Potential Effects	Probability value	Severity	Risk Assessment
Someone trips on an electric cord	Person might get hurt	1	9	9
Car falls off track at top	Car breaks, scout feels bad	3	8	24
Equipment gets bounced around in transport	Equipment breaks and must be fixed	4	2	8

However, because tape is so cheap, we should tape down the extension cords.

4. DISCUSSION

4.1. Shortcomings

Up until now, the Pinewood Derby was serving as a good example of how to do risk analyses. However, it fails to act as a good model for a commercial company, because a Pinewood Derby is run by volunteer parents.

In this paper we compared performance risks, we compared schedule risks, we compared cost risks, and we compared safety risks. But we did not compare performance risks *to* schedule risks, etc. We cannot make such comparisons, because their units for severity are different. However, combining all risks into one number, or evaluation, is exactly what we want to do. The most popular technique for combining different types of risk is to translate all risks into dollar amounts [Buede, 2000], as is done by insurance companies. But this would be hard to do for our Pinewood Derbies, because you cannot translate the time of volunteer parents into dollars. If we did a risk analysis for a commercial company, we would define utility functions, such as the number of people involved, the number of hours worked, their salaries, etc., and then we would use standard multiattribute utility techniques [Kirkwood, 1997]. However, in running a Pinewood Derby, is it better to have a lot of people involved or a few? Is it better for them to work a lot or a little? More is probably better (up to a point), because you are getting parents doing things with their children. Furthermore, salaries are irrelevant: Who should be preferred as a finish-line judge, a brain surgeon or a gardener? So the utility functions would be difficult to define for a Pinewood Derby.

If we were to do a risk analysis for a commercial firm, we would compute its return on investment (RoI):

$$RoI = \frac{\sum Savings}{Cost}.$$

Savings is computed using number of people, number of hours saved and salaries. Cost is computed using number of person hours devoted to the risk analysis and salaries. Hall [1999] says that RoIs of 10–20 are quite common.

4.2. Limitations

This paper did not cover risk management, which has five steps: (1) Select and tailor the risk management process, (2) identify risks, (3) analyze and assess risks, (4) perform risk abatement (or mitigation), and (5) track and evaluate risks [Boeing, 1997; Wideman, 1992]. This paper primarily discussed identification, analysis, assessment, and abatement of risk.

This paper only considered risk analyses for simple systems to be performed for and by every-day people. If we were to give advice to a governor or the President of the United States about complex systems, we would certainly want to use more complex techniques, such as those presented by Haimes [1998].

Arguably the best part of Haimes [1999] is the final paragraph where he reminds us that, according to the Heisenberg uncertainty principle, we cannot simultaneously measure position and velocity with high precision. Then he recalls the statement by Albert Einstein: "So far as the theorems of mathematics are about reality, they are not certain: so far are they are certain, they are not about reality." Haimes then applies these concepts to risk assessment to get:

> To the extent that risk assessment is precise,
> it is not real.
>
> To the extent that risk assessment is real,
> it is not precise. (*page 17*)

4.3. Unresolved Issues

In this paper we presented several techniques for doing risk analyses. We think that at least one of these techniques will work for most simple systems. Unfortunately, we cannot specify one technique that should be used for all situations.

There are also some issues that we have not resolved. For example, for some risks it will be possible to change the severity of failure but not the probability, for example, in accommodating for bad weather. For other risks it will be possible to change the probability of failure but not the severity, for example, in dealing with irate parents. But our techniques do not take this into consideration.

Often human decisions do not match the product of probability and severity. Many professional baseball batters wear a pad on their off-dominant elbow. The probability of getting hit by the ball on the elbow is low, so the estimated risk is low. However, they still use risk mitigation, because the severity is high.

4.4. Strategies for Handling Risk

There are many strategies for handling risk: (1) Eliminate the risk by changing requirements or using alternative solutions. (2) Transfer the risk, for example, by changing a function's implementation from software to hardware. (3) Prevent escalation of risk by continuous monitoring. (4) Consciously accept the risk and do nothing. (5) Share the risk by buying insurance. (6) Develop alternatives for critical items, e.g., cultivate alternative sources and prototype alternative designs. (7) Control the risk by driving down the probability or the severity. Controlling the risk, also known as risk mitigation, is the most common technique. It entails monitoring, tracking, planning, and implementing mitigating actions. This is primarily what we have done in this Pinewood Derby case study.

4.5. General Comments

Handling risk in the real world involves rational use of all these strategies. For instance, in later years we chose to accept the risk that a parent could be unhappy about a race outcome. This risk was assumed to be minimal with the parent not causing a real problem, so the severity was low. In designing the event, the design parameters tended to make a fair race with happy scouts and parents. This tended to minimize this risk, but no effort was expended explicitly for this purpose.

In cases where risks can be reduced, the level of effort is driven by a cost/benefit analysis. If we assign expected costs to risks, we can see our greatest vulnerabilities. But shear magnitude is not the whole story. We might also look at the sensitivity of total risk dollars to risk handling dollars. By calculating the cost of implementing a particular strategy versus the reduced expected cost of risk, we can make good decisions. In the example of this paper, we can see that spending hundreds of dollars on equipment boxes to protect computers that are not very likely to get broken gives a very poor risk cost reduction to spending ratio. If instead, we spend a couple dollars on duct tape for the extension cords to avoid human injury with large potential risk cost, the net savings is great. This also works comparing the cost of insurance against the costs of eliminating a risk. For instance, we may decide that the risk cost reduction of buying insurance does not compare to spending the same money for nets or pads to protect derby cars around elevated areas of track.

Finally the risk assigned to each possible cause is determined by the risk analysis. This analysis is not static though. Each time the system is changed, the risk analysis needs to be reviewed. As risks are eliminated or reduced, other risks will increase in relative importance. As a system is deployed, the actual risks will become better known. The analysis presented here is after a dozen years of data acquisition and is therefore far better than our initial analysis. In our first attempts, our risks were believed to be equipment failures. Inclement weather was not actually considered (we were in Tucson after all), and, as the system evolved with new instrumentation, the risks changed. The first derbies were not computer-based, so that extension cord trip hazards or computer failures were not relevant. As the system evolved, the risks evolved with it and so did the risk analysis.

ACKNOWLEDGMENT

We thank Mark Powell of the University of Idaho at Idaho Falls for thoughtful comments on the manuscript.

REFERENCES

B.S. Blanchard and W.J. Fabrycky, Systems engineering and analysis, Prentice-Hall, Upper Saddle River, NJ, 1998.

A.T. Bahill and W.J. Karnavas, Reducing state space search time, AI Expert B8(9) (1993), 28–35.

Boeing, Risk management process, Boeing Space and Defense Group, Kent, WA, 1997

D.M. Buede, The engineering design of systems: Models and methods, Wiley, New York, 2000.

W.L. Chapman, A.T. Bahill, and A.W. Wymore, Engineering modeling and design, CRC Press, Boca Raton, FL, 1992.

FMEA, Procedures for performing a failure mode, effects, and criticality analysis, Military Procedure MIL-P-1629, November 9, 1949.

Y.Y. Haimes, Risk modeling, assessment and management, Wiley, New York, 1998.

Y.Y. Haimes, "Risk management," Handbook of systems engineering and management, A.P. Sage and W.B. Rouse (Editors), Wiley, New York, 1999, pp. 137–174.

E.M. Hall, Risk management return on investment, Syst Eng 2 (1999), 177–180.

C.W. Kirkwood, Strategic decision making: Multiobjective decision analysis with spreadsheets, Duxbury Press, Belmont, CA, 1997.

C.W. Kirkwood, "Decision analysis," Handbook of systems engineering and management, A.P. Sage and W.B. Rouse (Editors), Wiley, New York, 1999, pp. 1119–1145.

McDonnell Douglas, McDonnell Douglas Risk Management Card, St. Louis, MO, 1997.

J.A. Moody, W.L. Chapman, F.D. Van Voorhees, and A.T. Bahill, Metrics and case studies for evaluating engineering designs, Prentice Hall, Upper Saddle River, NJ, 1997.

R.M. Wideman, Project and program risk management: A guide to managing project risks and opportunities, Project Management Institute, Upper Darby, PA, 1992.

A.W. Wymore, Model-based systems engineering, CRC Press, Boca Raton, FL, 1993.

A. Terry Bahill has been a Professor of Systems Engineering at the University of Arizona in Tucson since 1984. He received his Ph.D. in electrical engineering and computer science from the University of California, Berkeley, in 1975. His research interests are in the fields of systems engineering, modeling physiological systems, eye-hand-head coordination, validation of knowledge-based systems, system design, and systems engineering theory. He has tried to make the public appreciate engineering research by applying his scientific findings to the sport of baseball. He is the author of *Bioengineering: Biomedical, Medical, and Clinical Engineering* (Prentice-Hall, Englewood Cliffs, NJ, 1981), *Keep Your Eye on the Ball: Curve Balls, Knuckleballs, and Fallacies of Baseball* (with R.G. Watts) (Freeman, New York, 1990 and 2000), *Verifying and Validating Personal Computer-Based Expert Systems* (Prentice-Hall, Englewood Cliffs, NJ, 1991), *Linear Systems Theory* (with F. Szidarovszky) (CRC Press, Boca Raton, FL, 1992 and 1997), *Engineering Modeling and Design* (with W.L. Chapman and A.W. Wymore) (CRC Press, Boca Raton, FL, 1992), and *Metrics and Case Studies for Evaluating Engineering Designs* (with J.A. Moody, W.L. Chapman, and D.F. Van Voorhees) (Prentice Hall, Englewood Cliffs, NJ, 1997). He holds U.S. Patent Number 5,118,102 for the Bat Chooser, a system that computes the Ideal Bat Weight for individual baseball and softball batters. He is a registered professional engineer and Editor of the CRC Press Series on Systems Engineering. He is a Fellow of The Institute of Electrical and Electronics Engineers (IEEE) and of The International Council on Systems Engineering (INCOSE). He is chair of the INCOSE Fellows Selection Committee.

William J. Karnavas was born in Pittsburgh, Pennsylvania, May 30, 1963. He earned his B.S. from Carnegie Mellon University in 1984 and his M.S. from the University of Pittsburgh in 1986, both in electrical engineering. He then attended the University of Arizona and received his Ph.D. in Systems and Industrial Engineering in 1992. He did brief stint as a Fellow for Neurophysiology, in the Department of Neurosurgery, at the University of Pittsburgh Medical School. He has been at the IBM Transarc Lab for five years and is now a Senior Software Engineer testing distributed software. His research interests are testing large software systems, systems engineering theory, and computer-based bioinstrumentation.