# Problem Resolution Optimization

Don Porter
Senior Statistician
Motorola

***ABSTRACT***

*No matter how well we plan and execute software development, defects are generated and can escape to the customers. Failure to quickly resolve software problems leads to negative consequences for our customers and increases internal business costs. A quick deterministic method to prioritize problems and implement their solution helps to reduce cycle time and costs.*

*Achieving this goal requires several steps. The first is to determine a model that links problem resolution performance to institutional variables and problem characteristics. Statistical Design of Experiments (DOE) is a tool that provides data requirements for estimating the impacts of these variables on problem resolution. . Once data has been gathered the results of statistical analysis can be input into a mathematical optimization model to guide the organization.*

*This paper describes such an analysis. We used Analysis of Variance (ANOVA) to correlate problem resolution cycle time with three predictors, problem severity, problem complexity and engineer experience. The ANOVA output was used to parameterize a linear programming model using an Excel spreadsheet add-in tool.*

*There were several benefits to the project.*

- *Optimal allocation of problems to the engineering staff resulted in savings of time and money.*
- *A closer relationship between experienced and novice engineers.*
- *Knowledge of the necessary problem resolution effort provided a baseline for further problem process improvement.*

*This project applied two methods common to SEI level 5 organizations, Design of Experiments (DOE) and mathematical optimization. We also find application for the classical doctrine of comparative advantage developed by 18th century economist David Ricardo.*

## 1. Introduction

Even a well - executed software process introduces defects that impact both development and customer systems. In the cellular market, where a fair amount of saturation has occurred, system stability and reliability have become key issues with our customers. Thus it is critical for the problem resolution process to repair software defects quickly and effectively.

We found that our problem resolution process needed improvement. In order to effect improvements we determined problem areas and probable causes. We developed a plan and gathered data. The findings of our data analysis shaped a better process.

This paper is divided into two sections:

1. A brief statement of the difficulties presented by the old problem resolution process, our efforts to define a new one, and the successes that we realized.
2. A description of the quantitative tools that were employed, and the outputs that guided the development of our new process.

Process managers are introduced to some tools that can improve any process. People familiar with quantitative tools should find an interesting application herein.

## 2. The Problem Resolution Process

*The Old Problem Resolution Process*

We examined the way that software problems were assigned in the past. We found that our problem resolution process faced the following difficulties:

- There was no clear procedure for evaluating a given problem or estimating the effort required for resolution.
- Problem assignments were done on an ad hoc basis, with no group of engineers dedicated to problem resolution. Usually the next available engineer would be assigned the next available problem.
- Less experienced engineers were rarely assigned to fix a high impact problem, limiting their progress.

Consequently problem resolution entailed extra cycle time. Some problems were not getting fixed in a timely fashion, and new development was negatively affected when staff was reassigned to problem resolution. Other negative impacts included:

- Increased problem backlogs.
- New engineers received little training on customer issues.
- Operating in crisis mode, experienced engineers were called upon to fix problems.
- Major fixes were sometimes late.
- Testing was often duplicated.
- New releases were postponed.
- TL9000 metrics reflected poor problem resolution performance.

We wondered if there was an optimal way to assign problems to engineers. If there were, it would free more staff for work on other key issues. The critical determinants of problem resolution needed to be defined to discover that new process. It was apparent that these fell into two classes:

1. The skill level of engineers assigned to problem resolution, and
2. Factors that determine problem difficulty.

We needed to find suitable definitions to apply to these classes.

*Engineer Experience*

The first process change was to construct a team of engineers that would be responsible for evaluating, implementing and testing each problem fix. By doing this, we developed a pool of experience to draw on. The software engineer's experience levels were evaluated. We took into account their ability, total technical experience, and interest in specific areas. This led to a classification scheme to define **engineer experience**.

*Software Problem Characterization*

A review of the problem backlog evidenced the need for a better method of classifying problems. A scheme was developed to classify problems by their customer and system impacts. Customer impact was defined as **problem severity**, which we had already been assessing. Experienced engineers examined problem descriptions and symptoms to determine the extent to which it decreased system performance.

The system impact led to a definition of **problem complexity**. Problem complexity numbers accurately reflected the impact of the given problem to all potential customers. This definition of complexity is based on the difficulty of implementing the problem fix within our system. Problems overlapping several development groups would involve coordination concerns. By assigning a higher complexity value we could identify problems that would be more difficult to fix. Note that this definition of problem complexity is not related to traditional measures of software complexity.

*The New Problem Resolution Process*

We developed an efficient, deterministic method to prioritize problems. The old approach to problem assignment was replaced by assignment to an appropriate engineer based on an examination of each problem. Attempts are made to assign problems that interest the individual engineer, allowing them to broaden their technical expertise. Of course there would have to be no negative impact on the schedule.

The new process was an enormous success. Beyond schedule improvements, the engineer team was able to learn a lot in a short amount of time. Many of the newer engineers were able to gain satisfaction by resolving challenging issues they could handle. The entire team's knowledge base grew.

Our progress was due in no small part to a careful statistical and mathematical treatment of the data that we gathered. The quantitative tools and models we employed are explored in the next section.

## 3. Mathematical and Statistical Modeling

This section presents the quantitative aspects of the project in the following subsections:

1. Statistically designed experiments for data collection.
2. The statistical analysis method to determine model parameters.
3. A mathematical optimization model to improve the process.

*3.1 Statistically Designed Experiment*

We needed to estimate the impact of the problem characteristics and engineer experience on problem resolution cycle time. Historical data was available, but inadequate. For example, the most complex problems had consistently been assigned to the most skilled engineers. Therefore we did not know how less experienced engineers would perform on difficult problems.  This did not permit a full evaluation of our model.

Statistical Design of Experiments (DOE) is a set of techniques that help the analyst determine data requirements to estimate the parameters of any given model. There are several steps required to generate a statistically designed experiment:

1. Identify the response variable (or variables) to be modeled.

2. Determine the factors that influence that variable.
3. Determine the mathematical model by which the factors affect the response.
4. Determine appropriate factor settings.
5. Determine the number of "runs" required to estimate the model parameters. If possible, replicates are beneficial.

Step 1: Response Variables: Our goal was to understand problem resolution time. Two major components of cycle time were considered:
1. Problem assessment time.
2. Problem implementation time.

These two sub-processes employ distinct procedures and are typically performed by different individuals. Until a problem has been properly assessed its true severity and complexity are poorly understood.

Problem resolution cost was a second response and employed the same model as cycle time.

Step 2: Factors: The three predictor variables, problem severity, complexity and engineer experience were described above.

Step 3: Model: This required more thought. Some of the considerations are:
1. Do significant interactions occur? Gurus obviously have an advantage over novices for any type of problem. But perhaps that advantage is not constant. They may have an even *greater* advantage for some problems. This is an interaction effect.
2. Are there higher interactions? In this case a 3-factor interaction is the only higher order interaction available, given by Severity x Complexity x Experience. It was decided that such an interaction was highly unlikely.

The model that was selected included all possible 2-factor interactions. It is given by:

$T = S + C + E + (S \times C) + (S \times E) + (C \times E)$, where

T = Problem resolution time.
S = Problem Severity
C = Problem Complexity
E = Engineer Skill

Step 4: Factor Levels: The assigned factor values were:

| | | | |
|---|---|---|---|
| • Problem Severity: | High | Low | |
| • Problem Complexity: | Simple | Involved | Complex |
| • Engineer skill: | Novice | Experienced | Guru |

Step 5: Number of Runs: The product of the numbers of factor levels determines the total number of candidate experiments. With 2 x 3 x 3 = 18 runs one can estimate the model in part 3 with one extra run (because we don't need to estimate the 3-factor interaction). We decided that we could afford a total of 30 runs. Statistical software generated the data design.

In this paper 30 simulated data points have replaced the proprietary data from our research. The data are given in Table 1. Note that all possible combinations of factor levels are represented.

**Table 1: Project Data**

| Obs | Severity | Complexity | Engineer Experience | Assessment Time (Days) | Implement Time (Days) | Total Closure Time (Days) |
|---|---|---|---|---|---|---|
| 1 | Low | Simple | Novice | 2.5 | 6.0 | 8.5 |
| 2 | Low | Simple | Novice | 2.3 | 6.6 | 8.8 |
| 3 | Low | Simple | Experienced | 2.1 | 2.5 | 4.6 |
| 4 | Low | Simple | Guru | 0.6 | 1.3 | 1.9 |
| 5 | Low | Involved | Novice | 5.2 | 10.4 | 15.6 |
| 6 | Low | Involved | Experienced | 4.1 | 9.1 | 13.3 |
| 7 | Low | Involved | Experienced | 4.8 | 7.9 | 12.7 |
| 8 | Low | Involved | Guru | 3.2 | 8.5 | 11.8 |
| 9 | Low | Complex | Novice | 6.9 | 13.4 | 20.3 |
| 10 | Low | Complex | Novice | 6.7 | 13.4 | 20.1 |
| 11 | Low | Complex | Experienced | 4.9 | 11.4 | 16.3 |
| 12 | Low | Complex | Experienced | 5.2 | 10.7 | 16.0 |
| 13 | Low | Complex | Guru | 4.8 | 8.6 | 13.4 |
| 14 | Low | Complex | Guru | 4.4 | 9.2 | 13.6 |
| 15 | High | Simple | Novice | 5.1 | 11.0 | 16.0 |
| 16 | High | Simple | Novice | 5.4 | 10.0 | 15.5 |
| 17 | High | Simple | Experienced | 3.9 | 9.0 | 12.9 |
| 18 | High | Simple | Experienced | 3.8 | 8.9 | 12.7 |
| 19 | High | Simple | Guru | 3.4 | 7.2 | 10.6 |
| 20 | High | Simple | Guru | 3.5 | 5.6 | 9.1 |
| 21 | High | Involved | Novice | 5.2 | 12.1 | 17.3 |
| 22 | High | Involved | Novice | 5.4 | 10.6 | 16.0 |
| 23 | High | Involved | Experienced | 4.9 | 10.1 | 15.0 |
| 24 | High | Involved | Experienced | 4.9 | 10.2 | 15.0 |
| 25 | High | Involved | Guru | 4.6 | 9.9 | 14.5 |
| 26 | High | Involved | Guru | 5.2 | 9.4 | 14.5 |
| 27 | High | Complex | Novice | 7.7 | 14.7 | 22.4 |
| 28 | High | Complex | Experienced | 6.8 | 14.3 | 21.1 |
| 29 | High | Complex | Guru | 5.1 | 9.5 | 14.6 |
| 30 | High | Complex | Guru | 5.3 | 11.4 | 16.8 |

## *3.2 Statistical Method for Data Analysis*

Analysis of Variance (ANOVA) was used to analyze closure time components. This is the standard method to correlate a numerical response with qualitative predictors. The full 2-factor interaction model discussed above was fit first. Any terms that failed statistical tests for significance were eliminated to avoid over-fitting the model.

*Output*

The tables below give the results of the full model fitting. Each table gives the effect, the degrees of freedom, mean squares and statistics for parameter tests. The final column gives the confidence level 100% x (1 - pvalue) for each parameter test. The confidence level expresses our confidence that the term truly affects the level of the response variables. The root mean square error (rmse) is a measure of the unexplained variation of the response and the R-square the fraction of total response variation explained by the statistical model.

In Table 2 the effects are presented in order of importance.

**Table 2: ANOVA Results for Problem Assessment Time**

| Source | Degrees of Freedom | Type III Sum of Squares | Mean Square | F Value | Confidence Level |
|---|---|---|---|---|---|
| Complex | 2 | 44.43 | 22.21 | 137.9 | > 99.9% |
| Severity | 1 | 14.34 | 14.34 | 89.0 | > 99.9% |
| Experience | 2 | 13.97 | 6.99 | 43.4 | > 99.9% |
| Complex * Severity | 2 | 4.44 | 2.22 | 13.8 | > 99.9% |
| **Complex * Experience** | **4** | **1.53** | **0.38** | **2.4** | **92.8%** |
| Severity * Experience | 2 | 0.13 | 0.06 | 0.4 | 31.9% |

rmse = 0.401 R-Square = .9614

**Table 3: ANOVA Results for Problem Implementation Time**

| Source | Degrees of Freedom | Type III Sum of Squares | Mean Square | F Value | Confidence Level |
|---|---|---|---|---|---|
| Complex | 2 | 13.64 | 6.82 | 15.61 | > 99.9% |
| Severity | 1 | 11.46 | 11.46 | 26.24 | > 99.9% |
| Experience | 2 | 13.00 | 6.50 | 14.88 | > 99.9% |
| Complex * Severity | 2 | 10.48 | 5.24 | 12.00 | > 99.9% |
| Complex * Experience | 4 | 9.37 | 2.34 | 5.36 | 99.3% |
| **Severity * Experience** | **2** | **2.91** | **1.45** | **3.33** | **93.6%** |
| Assessment (Coefficient = -0.35) | 1 | 0.32 | 0.32 | 0.72 | 59.2% |

rmse = 0.661 R-Square = .9751

For both the assessment and implementation cycle time analyses the main effects of Complexity, Severity and Experience are significant at a very high confidence level. We concluded that the variables to predict cycle time were chosen well.

In both models at the Complexity by Severity interaction term was significant. The estimated coefficients for this interaction (not shown) were all negative. This indicates that the severity level and complexity level were "substitutes". That is, high complexity for a high severity problem would add less cycle time than it would for a low severity problem. The Complexity by Experience interaction was of borderline significance for assessment time, and quite significant for implementation time. The positive coefficients (not shown) indicated that a highly experienced engineer would spend proportionately more time on a complex problem than would an engineer of more limited experience.

Assessment time is not a significant predictor of implementation time for this data set. It is included anyway because it was an important predictor of implementation time in the original, proprietary data. It is suggested that problem assessment time always be tried out in implementation time models since it can improve the accuracy of implementation time predictions.

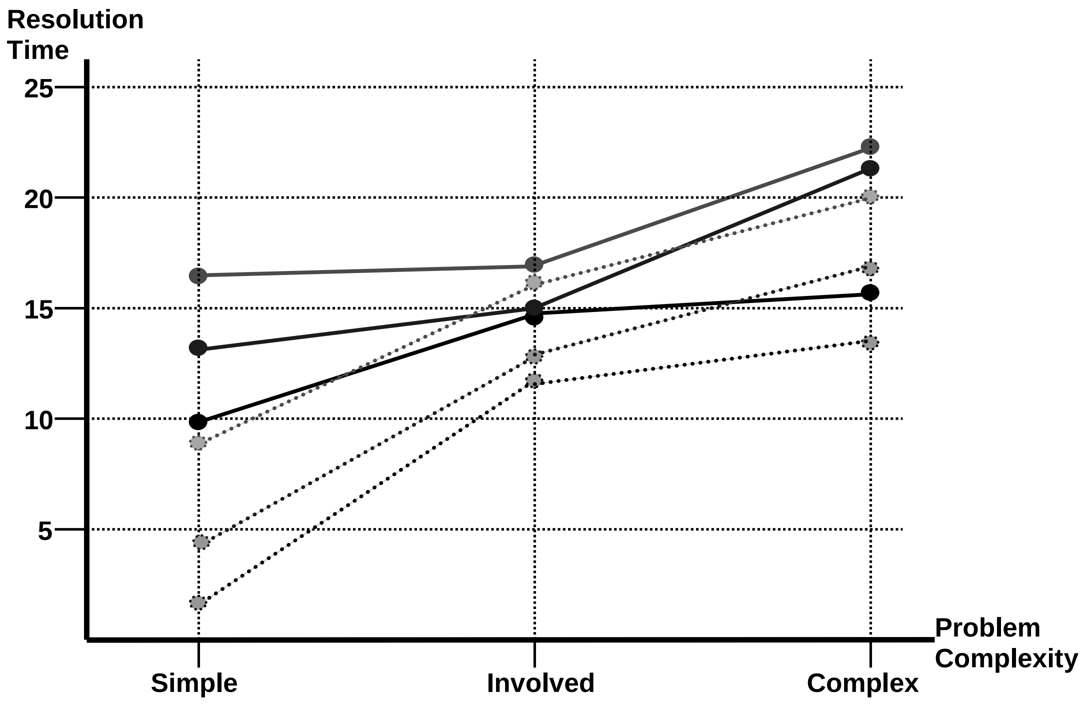
Table 4 contains the estimated mean problem closing times for each combination of severity, complexity and engineer skill. These means are used as input for the final optimization leg of the project.

**Table 4: Estimated Average Closing Times**

| Severity Level | Complexity Level | Engineer Experience | Assessment Mean (Days) | Implement Mean (Days) | Total Closure (Days) |
|---|---|---|---|---|---|
| High | Simple | Novice | 5.2 | 10.5 | 15.7 |
| | | Experienced | 3.9 | 9.0 | 12.9 |
| | | Guru | 3.4 | 6.4 | 9.8 |
| | Involved | Novice | 5.3 | 11.3 | 16.6 |
| | | Experienced | 4.9 | 10.1 | 15.0 |
| | | Guru | 4.9 | 9.7 | 14.6 |
| | Complex | Novice | 7.7 | 14.7 | 22.4 |
| | | Experienced | 6.9 | 14.3 | 21.2 |
| | | Guru | 5.2 | 10.5 | 15.7 |
| Low | Simple | Novice | 2.4 | 6.3 | 8.7 |
| | | Experienced | 2.1 | 2.5 | 4.6 |
| | | Guru | 0.6 | 1.3 | 1.9 |
| | Involved | Novice | 5.2 | 10.4 | 15.6 |
| | | Experienced | 4.5 | 8.5 | 13.0 |
| | | Guru | 3.3 | 8.5 | 11.8 |
| | Complex | Novice | 6.8 | 13.4 | 20.2 |
| | | Experienced | 5.1 | 11.1 | 16.2 |
| | | Guru | 4.6 | 8.9 | 13.5 |

These means are presented in an interaction plot in Figure 1.

**Figure 1: Interaction Plots for Total Problem Assessment Time**



In the interaction plot solid lines indicate high severity means and dotted lines the low severity means. Black lines represent means for gurus, blue for experienced engineers and red for novices. Note that the interaction plot is much “flatter” for high severity problems. The Complexity by Severity interaction is evidenced in this plot. Points are much closer together for complex problems than they are for simple problems, regardless of severity level. This indicates that the differences among engineer skill levels are accentuated most for the simple problems.

For problems of any severity level the increase in effort for gurus is greatest going from simple to involved problems, then flattens out for complex ones.  Perhaps their experience has given them greater facility in dealing with the integration aspects of complex problems.

The project was completed with the optimization leg, to which we now turn.

### *3.3 Mathematical Optimization Model*

The optimization problem is to minimize total resolution cycle time by assigning a given set of problems, defined by severity and complexity, to engineers of three skill levels.

The linear programming method was used to solve the optimization problem. There are 2 x 3 x 3 = 18 decision variables that represent each possible type of assignment. Each represents a number of problems of given severity and complexity assigned to engineers with one of the three skill levels. To make the notation more manageable the six classes of problems are indexed with numbers for problem complexity and engineer experience. The problems are labeled as follows:

$H_1$ = High Severity, Complexity 1 (simple) problems.
$H_2$ = High Severity, Complexity 2 (involved) problems.
$H_3$ = High Severity, Complexity 3 (complex) problems.
$L_1$ = Low Severity, Complexity 1 problems.
$L_2$ = Low Severity, Complexity 2 problems.
$L_3$ = Low Severity, Complexity 3 problems.

We also have the following resources.

$E^1$ = Number of Engineers of skill Level 1 (Novice)
$E^2$ = Number of Engineers of skill Level 2 (Experienced)
$E^3$ = Number of Engineers of skill Level 3 (Guru)

We denote the resource availability (in work days) by:

$T^1$ = Time availability of novice engineers
$T^2$ = Time availability of experienced engineers
$T^3$ = Time availability of guru engineers

Now we tie these definitions together. Let problems be indexed by severity i = H, L, complexity j=1,2,3, and engineer skill level k=1,2,3. Also identify the problem resolution average cycle times (estimated in the ANOVA) as:

$t_{ij}^k$ = time required to resolve a problem of severity i and complexity j by an engineer of skill level k.

And the number of problems be identified as:

$P_{ij}^k$ = Total number of problems of severity i and complexity j assigned to engineers of skill level k.

The formal statement of the optimization problem is then:
Minimize total problem resolution time

$$\begin{aligned}
\mathbf{T} = {} & (t_{H1}^1 \times P_{H1}^1) + (t_{H1}^2 \times P_{H1}^2) + (t_{H1}^3 \times P_{H1}^3) + \\
& (t_{H2}^1 \times P_{H2}^1) + (t_{H2}^2 \times P_{H2}^2) + (t_{H2}^3 \times P_{H2}^3) + \\
& (t_{H3}^1 \times P_{H3}^1) + (t_{H3}^2 \times P_{H3}^2) + (t_{H3}^3 \times P_{H3}^3) + \\
& (t_{L1}^1 \times P_{L1}^1) + (t_{L1}^2 \times P_{L1}^2) + (t_{L1}^3 \times P_{L1}^3) + \\
& (t_{L2}^1 \times P_{L2}^1) + (t_{L2}^2 \times P_{L2}^2) + (t_{L2}^3 \times P_{L2}^3) + \\
& (t_{L3}^1 \times P_{L3}^1) + (t_{L3}^2 \times P_{L3}^2) + (t_{L3}^3 \times P_{L3}^3)
\end{aligned}$$

Subject to the following constraints.

1. The total number of problems of each severity / complexity class to be resolved.

2. The endowment of engineers of each skill level.
3. Each of the 18 decision variables must be non-negative.

These sets of constraints are given algebraically as follows.

1. There are six equality constraints for the total number of problems:

$$\mathbf{P_{H1}^{1} + P_{H1}^{2} + P_{H1}^{3} = P_{H1}}$$
$$\mathbf{P_{H2}^{1} + P_{H2}^{2} + P_{H2}^{3} = P_{H2}}$$
$$\mathbf{P_{H3}^{1} + P_{H3}^{2} + P_{H3}^{3} = P_{H3}}$$
$$\mathbf{P_{L1}^{1} + P_{L1}^{2} + P_{L1}^{3} = P_{L1}}$$
$$\mathbf{P_{L2}^{1} + P_{L2}^{2} + P_{L2}^{3} = P_{L2}}$$
$$\mathbf{P_{L3}^{1} + P_{L3}^{2} + P_{L3}^{3} = P_{L3}}$$

2. There are three constraints for the total number of engineer staff-days:

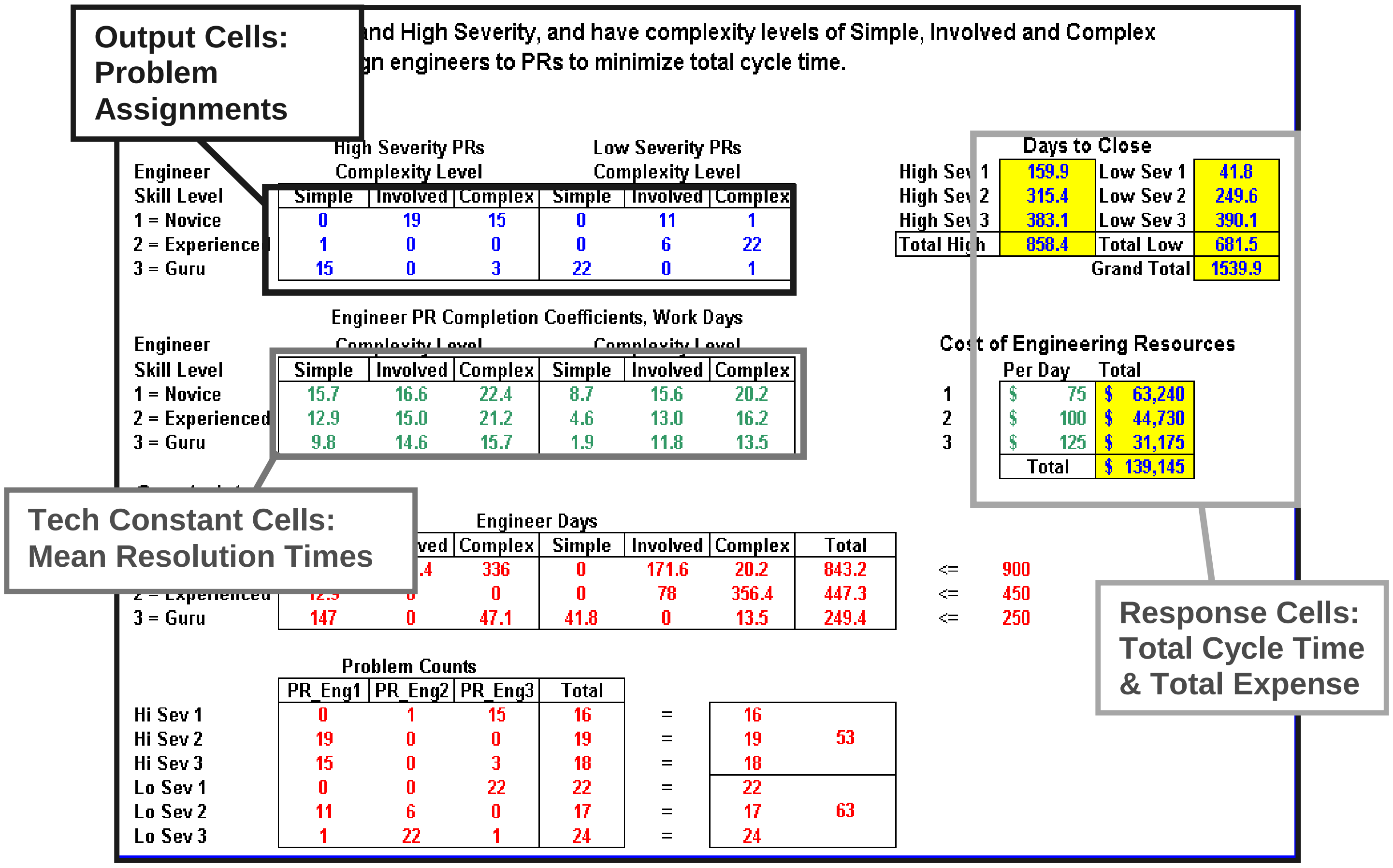
$$\mathbf{(T_{H1}^{1} \times P_{H1}^{1}) + (T_{H2}^{1} \times P_{H2}^{1}) + (T_{H3}^{1} \times P_{H3}^{1}) + (T_{L1}^{1} \times P_{L1}^{1}) + (T_{L2}^{1} \times P_{L2}^{1}) + (T_{L3}^{1} \times P_{L3}^{1}) \le T^{1}}$$
$$\mathbf{(T_{H1}^{2} \times P_{H1}^{2}) + (T_{H2}^{2} \times P_{H2}^{2}) + (T_{H3}^{2} \times P_{H3}^{2}) + (T_{L1}^{2} \times P_{L1}^{2}) + (T_{L2}^{2} \times P_{L2}^{2}) + (T_{L3}^{2} \times P_{L3}^{2}) \le T^{2}}$$
$$\mathbf{(T_{H1}^{3} \times P_{H1}^{3}) + (T_{H2}^{3} \times P_{H2}^{3}) + (T_{H3}^{3} \times P_{H3}^{3}) + (T_{L1}^{3} \times P_{L1}^{3}) + (T_{L2}^{3} \times P_{L2}^{3}) + (T_{L3}^{3} \times P_{L3}^{3}) \le T^{3}}$$

3. There are 18 non-negativity constraints:

$\mathbf{P_{ij}^{k}} \ge \mathbf{0}$; i = L, H; j = 1, 2, 3; k = 1, 2, 3

Optimization was performed using an Excel add-in program. The results appear in the following figures.

**Figure 2. Output from the Cycle Time Optimization Algorithm**


Output Cells: Problem Assignments
Tech Constant Cells: Mean Resolution Times
Response Cells: Total Cycle Time & Total Expense

nd High Severity, and have complexity levels of Simple, Involved and Complex
gn engineers to PRs to minimize total cycle time.

| | High Severity PRs | | | Low Severity PRs | | |
|---|---|---|---|---|---|---|
| Engineer Skill Level | Complexity Level | | | Complexity Level | | |
| | Simple | Involved | Complex | Simple | Involved | Complex |
| 1 = Novice | 0 | 19 | 15 | 0 | 11 | 1 |
| 2 = Experienced | 1 | 0 | 0 | 0 | 6 | 22 |
| 3 = Guru | 15 | 0 | 3 | 22 | 0 | 1 |

| Days to Close | | | |
|---|---|---|---|
| High Sev 1 | 159.9 | Low Sev 1 | 41.8 |
| High Sev 2 | 315.4 | Low Sev 2 | 249.6 |
| High Sev 3 | 383.1 | Low Sev 3 | 390.1 |
| Total High | 858.4 | Total Low | 681.5 |
| | | Grand Total | 1539.9 |

Engineer PR Completion Coefficients, Work Days

| Engineer Skill Level | Simple | Involved | Complex | Simple | Involved | Complex |
|---|---|---|---|---|---|---|
| 1 = Novice | 15.7 | 16.6 | 22.4 | 8.7 | 15.6 | 20.2 |
| 2 = Experienced | 12.9 | 15.0 | 21.2 | 4.6 | 13.0 | 16.2 |
| 3 = Guru | 9.8 | 14.6 | 15.7 | 1.9 | 11.8 | 13.5 |

Cost of Engineering Resources

| | Per Day | Total |
|---|---|---|
| 1 | $ 75 | $ 63,240 |
| 2 | $ 100 | $ 44,730 |
| 3 | $ 125 | $ 31,175 |
| | Total | $ 139,145 |

Engineer Days

| | | ved | Complex | Simple | Involved | Complex | Total | | |
|---|---|---|---|---|---|---|---|---|---|
| | | .4 | 336 | 0 | 171.6 | 20.2 | 843.2 | <= | 900 |
| 2 = Experienced | 12.9 | 0 | 0 | 0 | 78 | 356.4 | 447.3 | <= | 450 |
| 3 = Guru | 147 | 0 | 47.1 | 41.8 | 0 | 13.5 | 249.4 | <= | 250 |

Problem Counts

| | PR_Eng1 | PR_Eng2 | PR_Eng3 | Total | | | |
|---|---|---|---|---|---|---|---|
| Hi Sev 1 | 0 | 1 | 15 | 16 | = | 16 | |
| Hi Sev 2 | 19 | 0 | 0 | 19 | = | 19 | 53 |
| Hi Sev 3 | 15 | 0 | 3 | 18 | = | 18 | |
| Lo Sev 1 | 0 | 0 | 22 | 22 | = | 22 | |
| Lo Sev 2 | 11 | 6 | 0 | 17 | = | 17 | 63 |
| Lo Sev 3 | 1 | 22 | 1 | 24 | = | 24 | |

The red cells toward the bottom of figure 2 identify the problem constraints. The non-negativity constraints did not need to be explicitly stated because the optimization software assumed non-negativity unless directed otherwise. Figure 2 shows the initial endowments of 900 novice, 450 experienced, and 250 guru staff-days. There were 53 high severity problems, 16 of complexity level 1, 19 of level 2 and 18 of level 3. There were 22, 17, and 24 low severity problems of levels 1, 2, and 3, respectively.

The green box highlights the technological parameters (cycle time means). Note that these are the same values that appear in the ANOVA output (table 4).

The blue and orange boxes highlight the algorithm outputs. The blue values give the allocations of the various problems to the engineer skill levels. The orange box shows that cycle time is minimized at 1,539.9 total engineer days. Assuming costs of $75, $100 and $125 per work hour for the novice, experienced and guru skill levels, total problem resolution cost equals $139,145.

The program indicates some slack resources. 249.4 of the total allocation of 250 guru hours were used. There were 2.7 hours of experienced engineer resources left and more than one week of novice resources. They had time to arrange the celebration!

The optimal solution (in blue) is interesting because, contrary to expectations, the gurus are mostly assigned to low complexity problems, with some preference for low severity. The novices tended to be assigned to high severity, high complexity problems. This surprising result is due to the strong comparative advantage the gurus had in the less complex problems, requiring only two staff-days for the low severity ones!

According to the economic theory of comparative advantage developed by English economist David Ricardo, the crucial question we should ask is, "For which types of problems do the experienced engineers have the greatest comparative advantage?" The example in Table 5, which uses some of the means for low severity problems in our study, can illustrate the result.

**Table 5: Engineers and PRs**

| **Problem Type** | **Engineer Experience** | **Days to Close** |
|---|---|---|
| Simple | Novice | 8.7 |
| Simple | Guru | 1.9 |
| Complex | Novice | 15.7 |
| Complex | Guru | 9.8 |

Suppose we have 20 simple problems and 10 complex ones. These could be closed in either of the following ways:

Policy A: Give the gurus the complex PRs and the novices the simple ones.
Total staff-days to close complex PRs = 10*9.8 = 98 days
Total staff-days to close simple PRs = 20*8.7 = 174 days
Total staff-days = 98 + 174 = 272 days

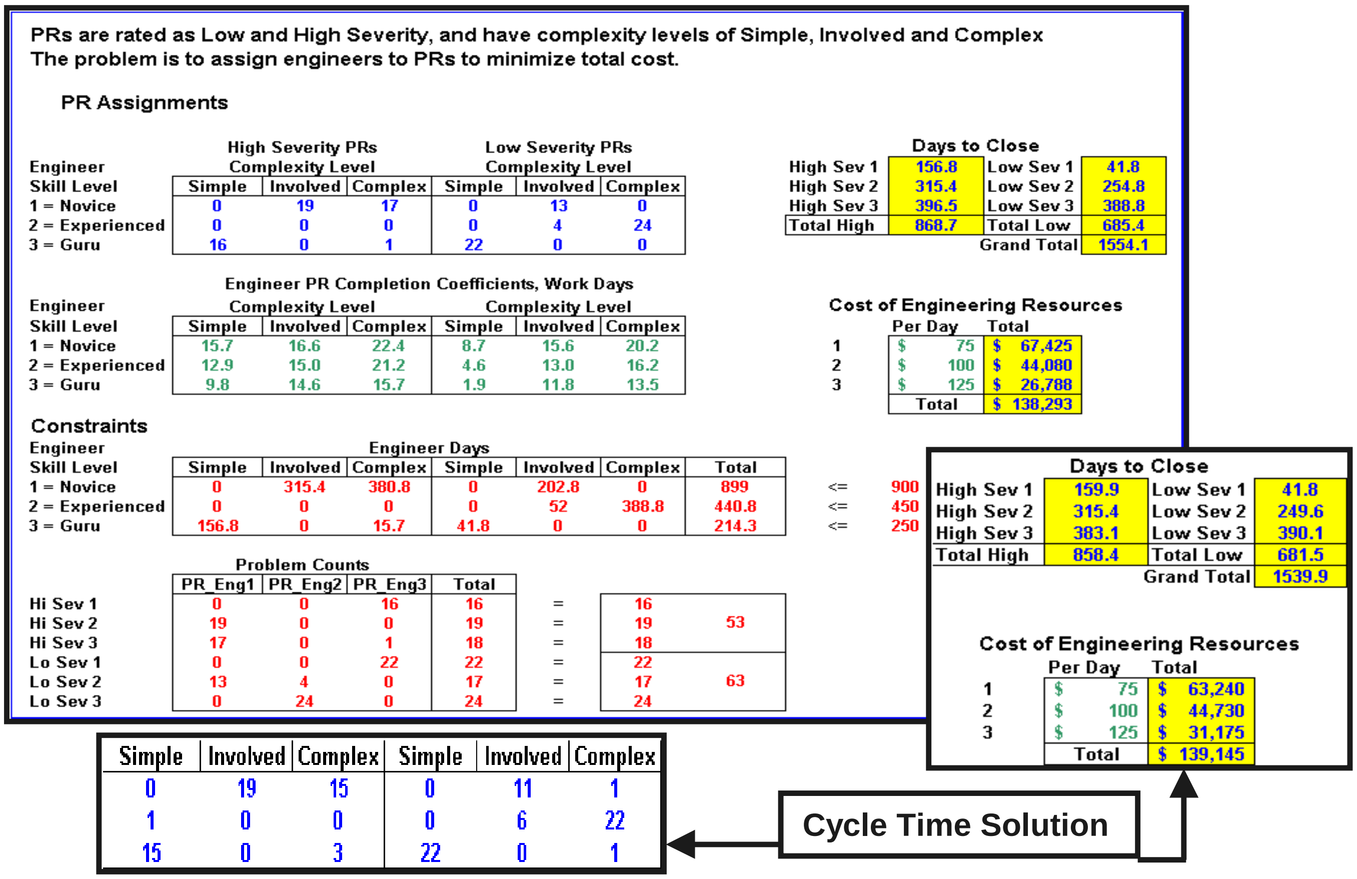
Policy B: Give the gurus the simple PRs and the novices the complex ones.
Total staff-days to close complex PRs = 10*15.7 = 157 days
Total staff-days to close simple PRs = 20*1.9 = 38 days
Total staff-days = 157 + 38 = 195 days

As this example shows, the time minimizing solution is to assign the complex problems to novice engineers (Policy B). Note that the gurus, compared to novices, have an advantage fixing the simple problems equal to 1.9 / 8.7 = .22, while their relative advantage for complex problems is 9.8 / 15.7 =

.62. The reader can verify that policy A would be better if the mix of simple to complex PRs were 10 to 20.

*Cost Minimization Model*
The algorithm was run again to determine the cost minimizing solution. The output from this exercise is found in figure 3.

**Figure 3. Output from the Cost Optimization Algorithm**



PRs are rated as Low and High Severity, and have complexity levels of Simple, Involved and Complex
The problem is to assign engineers to PRs to minimize total cost.

PR Assignments

| Engineer Skill Level | High Severity PRs Complexity Level: Simple | Involved | Complex | Low Severity PRs Complexity Level: Simple | Involved | Complex |
|---|---|---|---|---|---|---|
| 1 = Novice | 0 | 19 | 17 | 0 | 13 | 0 |
| 2 = Experienced | 0 | 0 | 0 | 0 | 4 | 24 |
| 3 = Guru | 16 | 0 | 1 | 22 | 0 | 0 |

Days to Close

| | | | |
|---|---|---|---|
| High Sev 1 | 156.8 | Low Sev 1 | 41.8 |
| High Sev 2 | 315.4 | Low Sev 2 | 254.8 |
| High Sev 3 | 396.5 | Low Sev 3 | 388.8 |
| Total High | 868.7 | Total Low | 685.4 |
| | | Grand Total | 1554.1 |

Engineer PR Completion Coefficients, Work Days

| Engineer Skill Level | Complexity Level: Simple | Involved | Complex | Complexity Level: Simple | Involved | Complex |
|---|---|---|---|---|---|---|
| 1 = Novice | 15.7 | 16.6 | 22.4 | 8.7 | 15.6 | 20.2 |
| 2 = Experienced | 12.9 | 15.0 | 21.2 | 4.6 | 13.0 | 16.2 |
| 3 = Guru | 9.8 | 14.6 | 15.7 | 1.9 | 11.8 | 13.5 |

Cost of Engineering Resources

| | Per Day | Total |
|---|---|---|
| 1 | $ 75 | $ 67,425 |
| 2 | $ 100 | $ 44,080 |
| 3 | $ 125 | $ 26,788 |
| | Total | $ 138,293 |

Constraints

Engineer Days

| Engineer Skill Level | Simple | Involved | Complex | Simple | Involved | Complex | Total | | |
|---|---|---|---|---|---|---|---|---|---|
| 1 = Novice | 0 | 315.4 | 380.8 | 0 | 202.8 | 0 | 899 | <= | 900 |
| 2 = Experienced | 0 | 0 | 0 | 0 | 52 | 388.8 | 440.8 | <= | 450 |
| 3 = Guru | 156.8 | 0 | 15.7 | 41.8 | 0 | 0 | 214.3 | <= | 250 |

Problem Counts

| | PR_Eng1 | PR_Eng2 | PR_Eng3 | Total | | | |
|---|---|---|---|---|---|---|---|
| Hi Sev 1 | 0 | 0 | 16 | 16 | = | 16 | |
| Hi Sev 2 | 19 | 0 | 0 | 19 | = | 19 | 53 |
| Hi Sev 3 | 17 | 0 | 1 | 18 | = | 18 | |
| Lo Sev 1 | 0 | 0 | 22 | 22 | = | 22 | |
| Lo Sev 2 | 13 | 4 | 0 | 17 | = | 17 | 63 |
| Lo Sev 3 | 0 | 24 | 0 | 24 | = | 24 | |

Days to Close

| | | | |
|---|---|---|---|
| High Sev 1 | 159.9 | Low Sev 1 | 41.8 |
| High Sev 2 | 315.4 | Low Sev 2 | 249.6 |
| High Sev 3 | 383.1 | Low Sev 3 | 390.1 |
| Total High | 858.4 | Total Low | 681.5 |
| | | Grand Total | 1539.9 |

Cost of Engineering Resources

| | Per Day | Total |
|---|---|---|
| 1 | $ 75 | $ 63,240 |
| 2 | $ 100 | $ 44,730 |
| 3 | $ 125 | $ 31,175 |
| | Total | $ 139,145 |

| Simple | Involved | Complex | Simple | Involved | Complex |
|---|---|---|---|---|---|
| 0 | 19 | 15 | 0 | 11 | 1 |
| 1 | 0 | 0 | 0 | 6 | 22 |
| 15 | 0 | 3 | 22 | 0 | 1 |

Cycle Time Solution

The total cost falls to $138,293 from the previous $139,145 for a savings of $852. The cycle time increases to 1,554.1 total hours from 1,539.9 hours for an increase of 14.2 hours. This was accomplished by substituting some of the novice resources that had previously been slack (using 899 of their hours compared to the previous 843.2 hours). Guru hours fell from 249.4 to 214.3, freeing up 35.1 hours for them to work on other projects. 6.5 hours of experienced engineers were also freed.

It can be seen that linear programming can be used to work out "what-if" scenarios, allowing project managers to see the consequences of choosing cost over schedule or vice-versa.