

Using Analytic Hierarchy Process (AHP) to Prioritize Software Defects

Dr. Richard Scotti and Robert H. Laurine Jr.

Department of Engineering Management, The George Washington University, Washington D.C.

Abstract: This report investigates the application of the Analytic Hierarchy Process (AHP) as a decision support tool to aid software managers in the prioritization of software defects (SODS). The effectiveness of an AHP SODS (prioritization) model for information management systems is discussed and illustrated by means of comparison between priorities set by the manager and those set by the model.

Today, software maintenance typically consumes more than half of all software development resources. Effective maintenance management is viewed as an important approach to controlling expanding software systems costs. One aspect of software maintenance management which may be improved upon is the area of software defect prioritization, which is the basis for maintenance planning and maintenance budgets.

Effectiveness of a SODS decision support tool was evaluated by means of a correlational study involving an operational information system. First, a sample group of 40 software defects was selected from a target population of 668 using a stratified random sampling technique. Then, two instruments were used to prioritize each sample software defect. The first was an after-the-fact instrument, consisting of the manual process currently used by an organization's software engineers to prioritize software defects. The second instrument was the AHP SODS model. Information system software managers representing domain experts for this study provided the technical and managerial criteria and subjective judgments used in building the AHP SODS model.

Early results are encouraging, and indicate that the AHP SODS model is an accurate and practical decision support tool for prioritization of the more severe (P2 level) software defects. Differences in the prioritization of the lower level (P3 through P5 level) software defects between the AHP SODS model and manual methods remain to be explained by continuing research in the Department of Engineering Management at The George Washington University.

Keywords: Analytic Hierarchy Process, Software Maintenance, Software Defects

1. Introduction

1.1. Statement of the Problem

The numbers of software defects for large scale software development efforts have become unmanageable as systems have become functionally and structurally more complex. Moreover, time and budget constraints necessitate choices by software managers on which defects to attack and when. Less experienced software managers may be expected to make less effective decisions regarding software defect prioritization, resulting in sub-optimal utilization of an organization's software maintenance resources. These factors have resulted in the need for a systematic prioritization methodology and for decision support tools to support both broadly and less experienced software managers. It is expected that application of such a methodology would be

helpful in improving management effectiveness in setting software defect priorities, in terms of both better decisions on how to use the resources, and reduced times and costs for the prioritization process itself.

1.2. Background

Software maintenance is composed of all the activities associated with a system after the system has been defined, developed, tested, deployed and accepted as operational. Typically, software maintenance consumes more than half of all system development resources. In fact, on some defense systems, 80% of the total software life cycle costs have been expended on maintenance [4]. Most of these costs may be attributed to the fact that much of the existing software was produced before the widespread use of structured techniques (e.g., software management metrics, Computer-Aided Software Engineering (CASE) tools). As a result, the importance of controlling these costs through effective maintenance management cannot be overstated.

An overview of the software maintenance process, adapted from Grady and Caswell [3], is shown in Figure 1. The process begins when a software problem is submitted by the customer. A customer request is followed by the establishment of an unclassified software defect report that is subsequently classified by a software manager. A defect is any product anomaly that may be of the form of an omission or imperfection in the existing software.

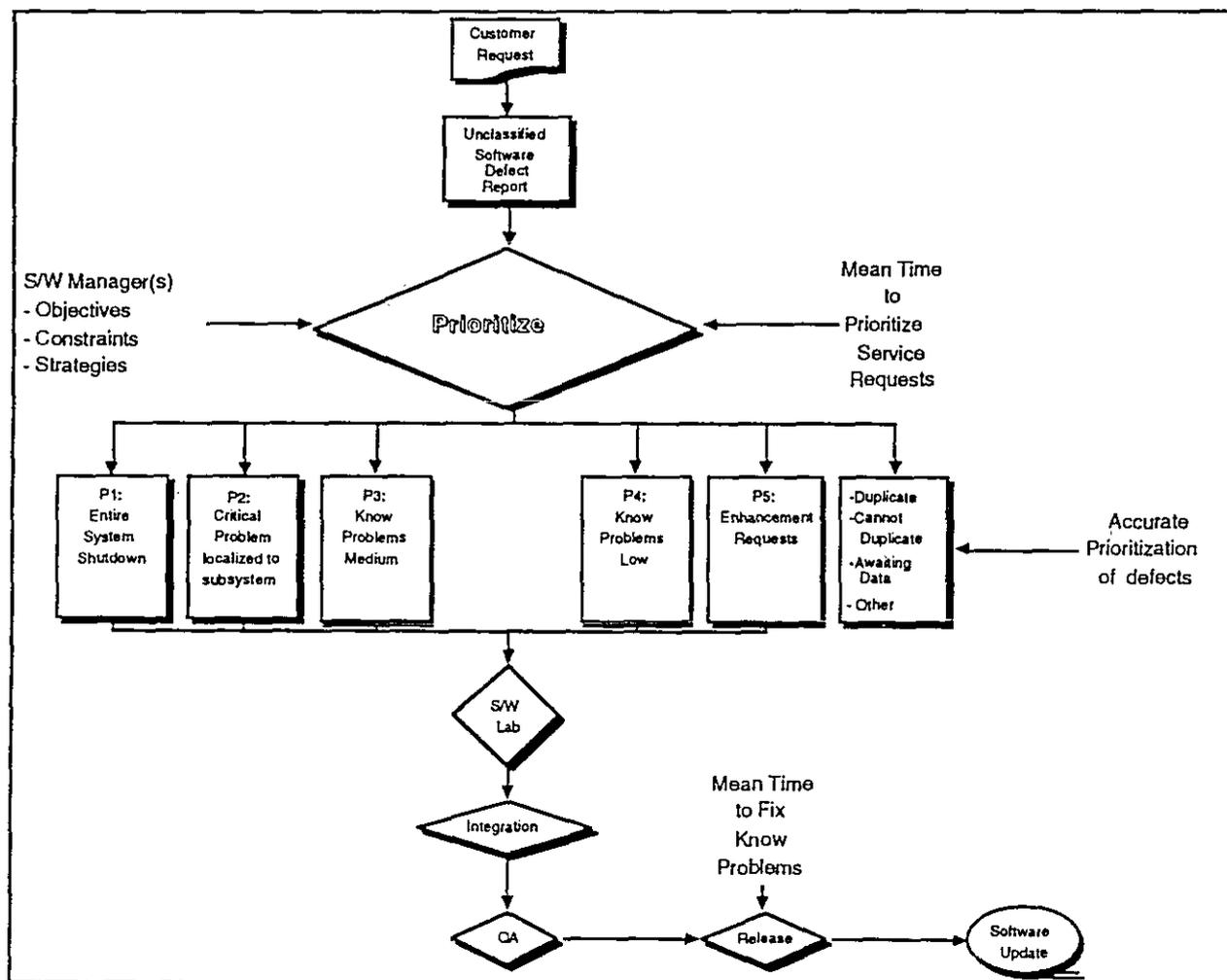


Figure 1. Software Maintenance Process

Managers operate on the stream of proposed defects to determine a set of approved changes. Priorities are established from both a management and technical perspective. From the management perspective, software managers set priorities under a variety of objectives, constraints, and strategies established by the maintenance organization. Typically, however, software managers tend to inflate their Initial priorities of their defects to ensure their acceptance for near-term repair. Later from a technical perspective, Final priorities are assessed on such factors as severity of the problem, effected software applications, and effected system outputs. These factors are considered, along with those of the maintenance management organization, in establishing a priority for each defect.

An example of a technical prioritization scheme suggested by Meredith [6] is as follows:

- (a) P1: a defect stopping all users from doing work (i.e., the entire system crashing)
- (b) P2: a defect stopping a user from doing work, or having appreciable risk of a workstation crash or of data loss
- (c) P3: a defect identified, however the activity can continue with an acceptable workaround, or an extremely low likelihood of a crash
- (d) P4: a minor problem (i.e., cosmetic problem)
- (e) P5: an enhancement of a philosophic issue (i.e., documentation)

Finally, working from the defect reports, the maintenance organization diagnoses and fixes the high priority problems which have been assigned to a maintenance effort. Software fixes must pass through a build process that involves the integration, quality assurance, and release of the final software updates, and be properly logged in the system configuration management system (CMS).

1.3. A multi-criteria decision problem

Formal control of the prioritization process is necessary to adequately address the software defects. This complex process must address the requirements of the various stakeholders which include the customer, user of the system, the software manager and upper management. Techniques, such as prioritizing software defects by "intuitive means" worked for smaller scale, less complex systems of the past. But these manual techniques are typically overwhelmed and inadequate for modern large scale development systems.

Prioritization enables the manager to receive the best return on investment associated with maintenance resources. For example, today's complex software systems typically experience defect rates of 15 to 18 per thousand lines of code [5]. It is not uncommon to see thousands of software defects for a large scale development effort. In addition, with average costs to fix a defect on the order of \$10K, general maintenance costs can escalate rapidly [1]. These realities support the following philosophy among software managers: "Since we can't fix them all, we have to be sure to fix the right ones."

Much of the recent software engineering literature identifies the detection and repair of software defects early in the software development life cycle as the most effective means to control software systems maintenance costs. Decision Support theory provides a variety of tools and models which may support a more effective approach to software defect management decisions. The present idea of using AHP as a tool to support prioritization of software defects is one such application and represents a new and important area of research.

2. Model Development

The factors of importance for establishing software defect priorities of concern to this study were identified by personal interviews of experienced software managers. Five (5) software managers, with extensive maintenance experience in the area of Information Management Systems, cooperated in this effort. The following factors are those they typically consider in selecting defects for resource expenditures:

2.1. Operational Impact

System operations are characterized by system configuration and functionality. A severe defect may bring the entire system to a stop. A less severe defect may result in the failure of one or several subsystems, or of one or more functions or workstations of the system. A much less severe defect may have a known, built-in workaround. Each of these cases, moreover, is characterized by their frequency of occurrence and the amount of resulting system downtime.

2.2. Production Impact

System production comprises the output or throughput of system. Production levels after a defect has occurred may be classified as high, medium or low depending on whether system output or throughput is reduced to zero, 50% or 90% of capacity, respectively.

2.3. Internal Users Affected

The internal users of a system are the personnel who operate on or work with the system in carrying out their job functions. User impacts may be considered high, medium or low when 100 or more, 50 to 100, or less than 50 users, respectively, are affected by the presence of a defect.

2.4. Ability to Fix

The assessment of the "fixability" of a defect is a complex matter which most often depends heavily upon the experience and expertise of the person making the assessment rather than any specific methodology. Highly experienced software managers, however, tend to characterize fixability of a defect by the reproducibility of the problem, by the level of difficulty associated with a software fix, and by whether or not the effectiveness of a fix can be tested.

2.5. Cost of the Fix

The cost of fixing a defect is typically estimated in terms of the expected number of lines of code required, and the complexity of the code to be modified.

2.6. Prior Scheduling

A defect may have already been identified and, subsequently, included in a future maintenance plan. Defects which have been previously scheduled for maintenance within 6 months should be given higher priority than those scheduled within 6 weeks. The highest priority should be given to defects which have not yet been scheduled.

These factors were expanded and consolidated into the hierarchical, multi-attribute decision tree model shown in Figure 2. Its purpose is to support a rational methodology for scoring software defects according to the strategy, "The higher the score, the higher the priority for fixing the defect."

Prioritization of S/W Defects

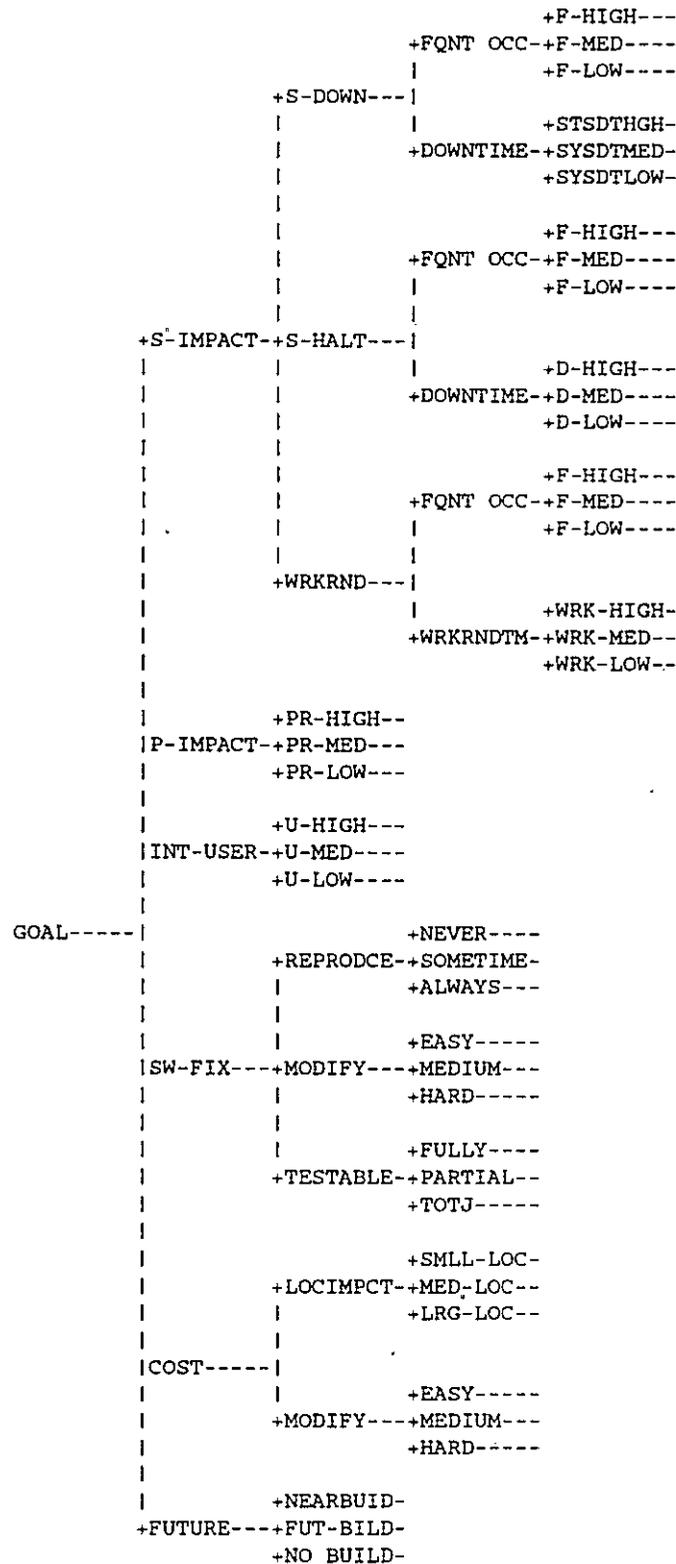


Figure 2. AHP SODS Model for prioritizing software defects

3. Evaluation

3.1. Judgments

The model is, of course, incomplete without weights at each of its branch points to indicate relative importance of each of the represented criteria and sub criteria. These were generated using an Analytical Hierarchical Process (AHP) in conjunction with the Expert Choice commercial software package [2]. The five (5) software managers were further questioned, this time by means of a standard paired-analysis questionnaire printed by Expert Choice, to obtain their subjective judgments. Results from the questionnaires were consolidated by averaging to yield the weights for the AHP software defects prioritization model (AHP SODS) shown below in Figure 3. This model was implemented in Expert Choice; it was structurally adjusted during development to eliminate biasing between nodes with unequal numbers of descendants.

Prioritization of S/W Defects

GOAL					
L 1.000					
G 1.000					

+-----+					
S-IMPACT	P-IMPACT	INT-USER	SW-FIX	COST	FUTURE
L 0.217	L 0.280	L 0.159	L 0.100	L 0.100	L 0.144
G 0.217	G 0.280	G 0.159	G 0.100	G 0.100	G 0.144

++S-DOWN	+-PR-HIGH	+-U-HIGH	++REPRODCE	++LOCIMPCT	+-NEARBUID
L 0.455	L 0.683	L 0.547	L 0.674	L 0.540	L 0.086
G 0.098	G 0.191	G 0.087	G 0.067	G 0.054	G 0.012
++S-HALT	+-PR-MED	+-U-MED	++MODIFY	++MODIFY	+-FUT-BILD
L 0.347	L 0.200	L 0.345	L 0.226	L 0.460	L 0.285
G 0.075	G 0.056	G 0.055	G 0.023	G 0.046	G 0.041
++WRKRND	+-PR-LOW	+-U-LOW	++TESTABLE		+-NO BUILD
L 0.199	L 0.117	L 0.109	L 0.101		L 0.628
G 0.043	G 0.033	G 0.017	G 0.010		G 0.090

Figure 3. The Goal of Prioritizing Software Defects

3.2. Preliminary Model Validation

Preliminary validation of the AHP SODS model was accomplished by the use of a set of five generic software defects which were developed by the five (5) software managers. Validation in this case is based on a comparison of the priorities generated by the AHP SODS model with those developed manually by the software managers. The first step of the validation procedure involved the design of a software defect for each (levels P1 through P5) of Meredith's [6] prioritization scheme. In this scheme P1 defects are the most important and P5 the least. These same generic defects were then independently scored, attribute by attribute, using the AHP SODS model. The AHP SODS model scores for each of the attributes are shown along with the model totals (far right hand column) in Table 1, which represents a ratings spreadsheet in Expert Choice. As anticipated, the model prioritization scores correlate well with the original priorities of the defects represented in the left most column in Table 1. Each of the generic defect scores reflects a priority which is

consistent with the original manual prioritizations. The P1 defect scored the highest score (0.957 out of 1.000), while the P5 defect scored the lowest (0.175 out of 1.000).

	S-IMPACT	S-IMPACT	S-IMPACT	S-IMPACT	S-IMPACT	S-IMPACT	P-IMPACT	INT-USER	SW-FIX	SW-FIX	SW-FIX	COST	COST	FUTURE	
	S-DOWN	S-DOWN	S-HALT	S-HALT	WRKGRND	WRKGRND			PEFFCOCE	MODIFY	TESTABLE	LOCIMPCT	MODIFY		
	FNTOCC	DOWNTIME	FNTOCC	DOWNTIME	FNTOCC	WRKGRNDTM									
ALTERNATIVES	0.0448	0.0547	0.0379	0.0379	0.0217	0.0217	0.261	0.1571	0.0675	0.0109	0.021	0.0543	0.0463	0.1632	TOTAL
P1 SYSTEM CRASH	F-HIGH	STSDHGH	F-HIGH	D-HIGH			PR-HIGH	U-HIGH	ALWAYS	EASY	FULLY	SMLL-LOC	EASY	NO BUILD	0.957
P2 W/S CRASH			F-HIGH	D-HIGH			PR-HIGH	U-HIGH	ALWAYS	EASY	FULLY	SMLL-LOC	EASY	NO BUILD	0.860
P3 WRONG FONT SIZE					F-LOW	WRK-LOW		U-MED	ALWAYS	EASY	FULLY	SMLL-LOC	EASY	NO BUILD	0.453
P4 COLOR SCREEN									ALWAYS	EASY	FULLY	SMLL-LOC	EASY	NO BUILD	0.344
P5 MOUSE SUPPORT								U-LOW						NO BUILD	0.175
#0 ????????????															

Table 1. Expert Choice ratings spreadsheet for five "engineered" software defects

4. Model Experimentation

4.1. The Research Question

The premise of the present study is that an AHP SODS model can be developed to improve a software manager's effectiveness and performance for prioritization of software defects. Improved effectiveness and performance should lead in turn to more effective utilization of software maintenance resources. For the purpose of the present research, this question was converted into the following Directional Hypothesis: Software managers who use AHP SODS as a decision support tool will more accurately and effectively prioritize software defects as compared to those who do not use it.

4.2. Research Approach

A set of 40 software defects was selected from a sample population of 668 for use with the following two instruments: A manual prioritization technique and the Expert Choice-based AHP SODS decision support tool. A correlation analysis was implemented to address the research hypothesis and to evaluate the effectiveness of the AHP SODS model.

4.3. Sample Selection

The sample population was comprised of all software problems encountered by a selected operational Information Management System since its date of initial operations. Each defect in the population was fully documented and under configuration control. Some were still in process of being corrected. The severity category distributions for the sample population are shown in Figure 4. P1 defects make up less than 1% of the population. P3 defects, those which affect customer functionality or system performance characteristics, account for approximately 67% of the defects in the population.

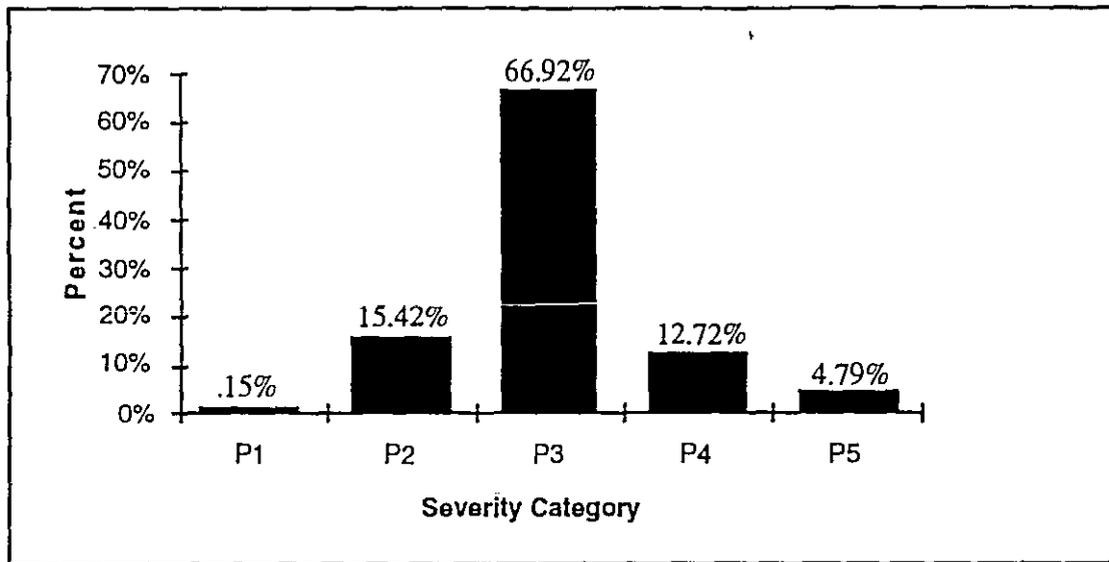


Figure 4. Distribution of Software Defects for the sample population

After elimination of the P1 defects from the sample because of extremely low frequency of occurrence, equal numbers for defects were selected for the remaining four severity categories using a stratified random sampling technique. Ten samples were selected for each category.

4.4. Study Instruments

The first of the two instruments used in this study consisted of the manual defect prioritization process currently used by software managers. The second was the Expert Choice-based AHP SODS model. Each of the five (5) software managers involved in the present study was well experienced in the use of the manual process. All were trained before the start of the study on the procedures for use of the AHP SODS model, which was configured as a ratings spreadsheet.

4.5. Test and Procedures

The 40 sample software defects were uniquely numbered in order to conduct an independent evaluation using the two test instruments. For the manual method, historical prioritization data was extracted from the CMS. This data included the Initial Priority and the Final Priority for each of the sample software defects. The Final Priority also served as the control data.

For the AHP SODS method, prioritization was conducted using a rating spreadsheet. For this case, the software managers were provided technical information extracted from the CMS to support the evaluation of each software defect. Each software manager was presented with an Expert Choice ratings spreadsheet, similar to that shown in Table 1, for the prioritization exercises. The spreadsheet represented the 40 software defect alternatives as its rows and the software engineering criteria used to evaluate the defects as its columns. The software manager simply prioritized each software defect under each criterion by selecting the appropriate intensity for it. The numerical scores from each cell are automatically added across all columns by Expert Choice to develop a total score for each alternative (row).

Table 2 illustrates the variables of interest considered in the correlational study, by defect number, including; (1) the AHP SODS priority, (2) the Initial Priority, and (3) the Final Priority. The AHP SODS priority was correlated against both the Initial Priority and the Final Priority. The correlation between the Initial Priority and the Final Priority was also calculated for completeness.

Defect Number	SODS Priority	Initial Priority	Final Priority
1	.890	1	2
2	.689	2	2
3	.438	3	3
4	.110	3	4
.	.	.	.

Table 2. Sample Priority Data by Defect Number

4.6. Results

First, the correlation between the Initial and Final priority resulted in a high positive value. Thus with an alpha value of .05 and $n=30$ (i.e., limit of critical values) the test statistic was greater than (.364) as shown in Table 3. A strong correlation relationship using the Spearman rho and Pearson r correlation methods was expected because the proposed priority is considered in the process of determining the Final priority.

Correlation Summary Table	Spearman rho	Pearson r
Proposed Priority - Final Priority	.664	.667
SODS Priority - Proposed Priority	-.584	-.645
SODS Priority - Final Priority	-.586	-.657
N=40, P<.05		

Table 3. Summary Correlation Summary Matrix

AHP SODS Priorities were correlated with both the Initial (proposed) Priorities and the Final Priorities. Spearman rho and Pearson r both indicated strong (negative) correlations. The negative correlations are due to the inverted scoring systems used by the two different methods. In addition, the magnitudes of the Pearson r correlations were close to those calculated between the Initial Priorities and the Final Priorities. These results support the position that AHP SODS Priorities nearly duplicated those developed by the manual prioritization procedures. Both Spearman rho and Pearson r approaches indicated strong correlational relationships between the different prioritization schemes under study.

Further investigation of the performance of the AHP SODS model was conducted using ANOVA to test the hypothesis that means from each of the software defect categories were equal. Final priorities were sorted by severity with their associated AHP SODS ratings. An F ratio of 13.81 was determined using 3 and 36 degrees of freedom, as shown in Table 4. Using an alpha of .05 the value of F required for significance is 2.92. These results indicate that the means of the software defect categories, those based on AHP SOD and those based on the manual method, were not equal.

SOURCE	DF	SS	MS	F	p
F-Prior	3	0.6409	0.2136	13.81	0.000
ERROR	36	0.5570	0.0155		
TOTAL	39	1.1980			

Table 4. ANOVA Results

Finally, an analysis was conducted to determine which of the software defect category means differed for the two methods being tested. The first step involved analyzing the software defect category means from ANOVA. These results are shown in Table 5. They suggest that the means of the P2 level defects are different from the remaining defects P3 through P5, while the remaining defects are not significantly different between themselves.

LEVEL	N	MEAN	STDEV
P2	10	0.6429	0.1414
P3	10	0.3970	0.1289
P4	10	0.3631	0.0966
P5	10	0.3151	0.1262

Table 5. ANOVA Descriptive Summary

Overall these results clearly support the adequacy of the AHP SODS model as a prioritization tool for P2 level software defects. Results for the remaining P3 through P5 level defects are ambiguous and suggest the need for further model development.

5. Discussion, Conclusions and Recommendations

This study was undertaken to test the possibility that an AHP SODS model might be developed to support more effective and accurate software defect prioritization by software managers. The application of such a model would enhance a software maintenance organization's ability to better use its resources, including the time spent by software managers in manually developing software defect priorities.

Based on the statistical analysis of a small sample of software defects, the model performed reasonably well in prioritizing the software defects. Results indicate that the most severe P2 level defects were accurately scored receiving the highest AHP SODS scores, while the model was somewhat less accurate in scoring for the remaining P3 through P5 level defects. Historically, the manual method has also had the same difficulty in prioritizing the P3 through P5 software defect categories. The reason is attributed to the fact the Information Management Organization has not yet been able to adequately characterize those defects.

Continuing research in the Department of Engineering at The George Washington University will focus on the following areas:

- (a) Further refinement of the current model by adjustment of the intensities associated with some of the (sub)criteria. This adjustment would be based on further AHP analysis using additional software managers with relevant backgrounds and experiences. A more comprehensive statistical analysis could also be conducted to determine which criteria are the principle "drivers" of the discrepancies between the two methods.
- (b) Further characterization of lower level software defect (sub)criteria. These localized criteria need to be further refined through additional research with the software managers. They may take the form of how visible the defect is to the user. For example, defects regarding documentation would have a small visibility to most users and hence be non-important to most users. While defects that involve terminal displays or other interfaces would have a greater visibility to the user.

(c) Extension of the evaluation process to other less experienced personnel within the organization to support the extension of the model for use by junior software managers. This extension could reduce the number of software defects to be evaluated by upper management, thereby reducing costs and freeing the top managers for more important project management tasks.

(d) Identification and use of real world information from other large scale software development projects that have entered into the maintenance phase for further validation of AHP SODS.

Future research should also focus on extending the application of AHP SODS to address other phases of large scale development efforts. For example, criteria could be modified to prioritize software defects over the code and unit test or integration phases of a system before delivery. In addition, research should examine other measures of effectiveness such as the mean time to prioritize software defects. The application of the AHP SODS decision support tool may significantly reduce the overall prioritization mean time.

References

[1] Bush, Marilyn, "Improving Software Quality: The Use of Formal Inspections at the Jet Propulsion Laboratory", *IEEE Experience Report*, (1990) 196-198.

[2] Forman, Ernest H., *Expert Choice Decision Support Software*, Pittsburgh: Decision Support Software (1991).

[3] Grady, Robert B., and Deborah L. Caswell. *Software Metrics: Establishing a Company-Wide Program*, Englewood Cliffs: Prentice-Hall (1987).

[4] Jensen, Randall W., Charles C. Tonis *Software Engineering*. Englewood Cliffs: Prentice-Hall (1990).

[5] Keene, Samuel J., "Software Reliability Concepts", *IEEE Annual Reliability and Maintainability Symposium* (1992) 1-21.

[6] Meredith, Denis C., "Software Metrics across a System's Lifecycle", *U.S. Professional Development Institute*. (1993) 27-37.