2008

COST ANALYSIS OF REQUIREMENTS, DESIGN, AND CODE REVIEWS AND HOW THIS IMPACTS THE COST OF QUALITY

Mark W. Huston
The University of Montana

COST ANALYSIS OF REQUIREMENTS, DESIGN, AND CODE REVIEWS AND HOW THIS IMPACTS THE COST OF QUALITY

By

Mark William Huston

Bachelor Science in Computer Science, University of Montana, Missoula, MT, 2004

Thesis

presented in partial fulfillment of the requirements
for the degree of

Master of Science
in Computer Science

The University of Montana
Missoula, MT

Autumn 2008

Approved by:

Dr. David A. Strobel, Dean
Graduate School

Dr. Joel Henry - Chairperson
Computer Science

Dr. Yolanda Reimer
Computer Science

Dr. Shawn Clouse
Business Administration

Huston, Mark, M.A., Dec 2008 Computer Science

Cost Analysis of Requirements, Design, and Code reviews and how this impacts the cost of quality

Chairperson: Dr. Joel Henry

This thesis examines the difference between accepted theoretical and real world return on investment of requirements, design, and code reviews. The differences have a significant impact on the cost of quality. The goal of this thesis is to examine the differences between two data sets (one derived from widely accepted software principles and one derived from real-world data) and draw conclusions about the effectiveness (cost vs. increase defect detection) of reviews based upon these analyses. This thesis will compare accepted relationships pertaining to cost per defect and overall project cost against actual data from a real world project in order to discover any significant differences. This research will also develop a cost estimation tool than may be used in future research to further develop the ideas and conclusions from this thesis. This author speculates that the cost benefit of reviews will decrease as the amount of time devoted to the review increases. This is contradictory to the accepted project management literature currently in wide use today.

Acknowledgements

I would like to thank Joel Henry, Ph.D., Yolanda Reimer, Ph.D., and Shawn Clouse, Ph.D. for their valued input and support. Special thanks to my wife Jani, sons Lane and Anthony, and daughter, Tessa.

Table of Contents

Table of Figures

Glossary

NoD = Number of Developers

CpH = Cost per Hour

NoT = Number of Testers

OhC = Overhead Costs

HoR = Hours of Review

NoHtFD = Number of hours to fix defects

CpHfDev = Cost per hour for developer

CpHfTest = Cost per hour for tester

NoDF = Number of defects found

CoR = Cost of Review

CtFDFdR = Cost to fix defects found during review

TCoaP = Total cost of all phases

NoHtFixDiCP = Number of hours to fix defects found in coding phase

NoHtTestDiCP = Number of hours to test defects found and fixed in coding phase

TD = Total defects

RH = Review Hours

CpHTesting = Cost per hour to test defect correction

CpHFix = Cost per hour to fix defects

NoHtTD = Number of hours to test defects

RD = Remaining defects

TH = Testing Hours

ND = Number of defect

Chapter 1 – Intent

The purpose of this paper is to examine the cost/benefits tradeoff of performing design, requirements, and code reviews in the development of software products. It has been theorized, and widely accepted, that performing requirement, design and code reviews will lower the overall cost of a project at a constant linear rate no matter how much effort is invested in reviews. McConnell states that "recent studies have shown conclusively that inspections are cheaper than testing" (1993, p. 565) Many software engineers believe that 'the use of inspections has improved productivity and product quality" (Fagan M. E., 1976, p. 258). However, the issues of cost and defect detection efficiency have not been completely factored into the interrelationship of reviews at various levels of effort performed throughout a software project. This thesis compares accepted relationships to actual data in order to discover any significant differences.

This research will show companies where they can efficiently spend their technology dollars and how much investment to make in reviews at these points in the development process. Many businesses struggle with allocation of technology funds and this thesis will assist them by showing IT professionals, and others who authorize project funds, where they can get the most bang for their buck. This assistance will further develop the software engineering field as software engineers strive to make it a more robust and definitive science.

This research will develop a cost estimation and data generation tool and use this tool to create models and then analyze the cost and benefits, with respect to time and

money, of requirements, design, and coding reviews. The time associated with each phase will be calculated and noted. The cost of each phase will be calculated along with the amount of time for each phase. Then these numbers will be compared with a real world project to determine the relationship between the theoretical models and the real world application of the review processes.

Chapter 2 – Hypothesis

The accepted principle examined here is: performing requirements, design, and code reviews will lower the overall cost of a project at a constant rate across the entire data space (hours invested in reviews), and improve the overall quality of the project (increase the defects found prior to testing). Alternatively, this thesis will analyze real world data from multiple projects that will show one organization's increasing investment in time and effort expended in reviews. This analysis will focus on the return on investment of the increased effort when preparation costs and reduced defect detection rates per hour are factored into the analysis. The initial costs of a review will cause the costs to go up but the overall net effect will be to reduce cost at a significantly slower rate. As the amount of time spent on reviews goes up the number of defects detected will increase but at a diminishing rate as well. Overall product quality will improve as more latent defects in the delivered product are removed, but again at a diminished rate per hour invested while review preparation time increases nonlinearly making these reviews more expensive per defect found.

Chapter 3 - Process

The theoretical analysis was conducted using models widely accepted in the field of Software Engineering, (Henry, 2003; McConnell, 1993; Pressman, 2001; Somerville, 2001) which were then compared with data from a real project conducted by a large aerospace company. The data from the aerospace company is proprietary so none of the actual data can be shown, but rather the overall costs and defect rates can be shown and compared. The theoretical analysis was conducted using ten different data sets. The four measurements were:

- Cost per phase in dollars
    - Requirements Review Cost
    - Design Review Cost
    - Code Review Cost
- Total Project Cost in dollars
    - Total Cost with Reviews
    - Total Savings with Reviews
    - Total Cost without Reviews
- Total Hours of Each Phase
    - Requirements Review hours
    - Design Review hours
    - Code Review hours
- Total Hours of Project

- Total Hours of Project with Reviews
- Total Hours of Project without Reviews

The results of the theoretical analyses were analyzed using the percentage of defects found. Theoretical analyses were conducted using 100 defects and 1000 defects initially present in each phase of the project. The number of hours for conducting the reviews was adjusted for each experiment and the results were recorded. The number of defects was kept constant for each theoretical analysis while the percentage of defects found was modified and recorded.

Chapter 4 – Tool

The tool used to analyze the data was developed using accepted best practices of software engineering. The equation for to determine the cost of each review was calculated using the following formula:

$$((NoD * CpH) + (NoT * CpH) + OC) * HoR))$$

NoD = Number of Developers
CpH = Cost per Hour
NoT = Number of Testers
OC = Overhead Costs
HoR = Hours of Review

Once the cost for the review was determined the next step was to determine the cost to fix the defects found during the review, a cost rarely considered in theoretical models. This number, the defects found, was the variable that changed for each theoretical analysis. This was calculated with the following formula:

$$(((NoHtFD * CpHfDev) + (NoHtTD * CpHfTest)) * NoDF)$$

NoHtFD = Number of Hours to Fix Defects
CpHfDev = Cost per hour for developers
NoHtTD = Number of Hours to Test Defects
CpHfTest = Cost per hour for Testers
NoDF = Number of Defects Found

This calculation determined the amount of money to fix the defects found during the inspection. The final calculation determined the total cost for each phase. This calculation was:

$$CoR + CtFDFdR$$

CoR = Cost of Review
CtFDFdR = Cost to Fix Defects Found during Review

These calculations were used to determine the final cost of each phase of the development process. The last calculations were to determine the final overall cost of the development of the software. Some assumptions were made at this point. One was that 100 percent of the defects were corrected. The other was that the earlier in the process

the defect was found; the less time it took to fix and test. "Although the benefits of inspections have been well studied, their costs are often justified by simply observing that the longer a defect remains in a system, the more expensive it is to repair, and therefore the future cost of fixing defects is greater than the present cost of finding them (Porter, Siy, Toman, & Vota, 1995, p. 92). The calculation is:

$$((TCoaP) * (CpHfDev * NoHtFixDiCP)) + ((CpHfTest * NoHtTestDiCP)) * RD)$$

TCoaP = Total Cost of all Phases
CpHfDev = Cost per Hour for Developer
NoHtFixDiCP = Number of Hours to Fix Defects in Coding Phase
CpHfTest = Cost per Hour for Tester
NoHtTestDiCP = Number of Hours to Test Defects in Coding Phase
RD = Review Hours

The calculation used to determine the total cost of development without any reviews was:

$$(((CpHfDev * NoHtFixDiCP) + (CpHfTest * NoHtTestDiCP)) * TD)$$

CpHfDev = Cost per Hour for Developer
NoHtFixDiCP = Number of Hours to Fix Defects found in Coding Phase
CpHfTest = Cost per Hour for Testers
NoHtTestDiCP = Number of Hours to Test Defects found in Coding Phase
TD = Total Defects

The final calculation was determined by taking the total amount without reviews minus the total amount with reviews.

The tool also calculated the number of person hours required to complete the project. The number of hours for a review was determined by using the following formula:

$$((RH * CpH) + (TH * CpH))$$

RH = Review Hours
CpH = Cost per Hour
TH = Total Hours

Each phase was calculated using this formula, and then the results were summed to determine the entire cost of the review process. The total cost was determined by taking this amount and adding it to the cost to correct the remaining errors. This was determined by calculating the cost per hour + remaining hours and adding to the total cost of the reviews.

The final calculation was obtained using the following calculation:

$$((CpHFix * ND) + (CpHTest * ND))$$

CpHFix = Cost per Hour to Fix defects
ND = Number or Defects
CpHTest = Cost per Hour to Test defects correction

The assumption was made that developers will correct every defect and an average amount of effort, or every defect took the same amount of effort, was required to repair each defect. While this works in a theoretical model, it does not correlate very well to an actual production environment where some defects require little effort and a few defects require a large number of person hours. Therefore we can consider this standard effort as the average across all defect repair times.

This tool does not include variables for the quality or experience of the reviewer, the preparation time associated with the reviews, any change in the frequency of finding the defects, and finally the tool does not take into account any changes in the error detection rate. However, it should be noted that few specific measurement approaches for these factors exist, and none of these approaches are accepted by academics or commercial organizations. The tool utilizes the currently accepted methods for defect detection which average these factors into an overall defect detection rate for reviews (Ahern, Clouse, and Turner, 2008).

Chapter 5 – Analysis

The theoretical analysis was conducted using several different scenarios. Throughout the analysis the cost for developers and testers was constant, $50.00 per hour and $25.00 per hour respectively. According to 2008-09 US Department of Labor, Occupational Outlook handbook, the "medium annual earnings of wage-and –salary computer applications software engineers were $79,790" (Buereau of Labor Statistics, U.S. Department of Labor, 2007). This average was increased by 25% to account for workers compensation insurance, social security taxes, and other expenses that an employer pays to have someone on the payroll for a total of $99,737 annually. Dividing this annual cost by 2080 hours, the number of hours worked based upon a 40 hour work week, resulted in an average hourly rate to employ a Software Engineer of $47.95 per hour. To simplify the analysis' results, the hourly rate was rounded up to $50.00 dollars per hour. The tester rate was arbitrarily decided to be ½ the rate of the software engineer. The number of developers and number of testers for the reviews was constant at 10 developers and 4 testers. The hours to correct a defect were set at 4, 8, 16 for the requirements phase, design phase, and coding phase respectively, and the hours to test a corrected defect were 2, 4, and 8. While the numbers were arbitrarily set, the ratio was not, M.E. Fagan points out that errors corrected early in the process are "10 to 100 times less expensive than if it is done in the last half of the process" (1976, p. 262). The number of hours to correct a defect found during the deployment/maintenance phase was 32 hours with 16 hours allocated to test the corrected defect.

The number of review hours were changed for each test but remained constant throughout each scenario. The number of defects found was changed and the resulting information was noted. The results were captured for 0 – 100 percent of defects found in 10% increments. These numbers were then placed into a graph and analyzed. In several of the theoretical analyses the data points were changed from the constant 10% increment to a ½X ratio and another analysis of -½X+100 ratio. In the ½X ratio, the data points were set to 25, 50, and 100 in the requirements, design, and coding phases. The subsequent data points were incremented by 2X or for example data point 2 would be 50, 100, and 200, and data point 3 would be 75, 150, and 300. This same pattern was repeated for the next 7 data points. Figure 1 below, shows this ratio in a graphical representation.

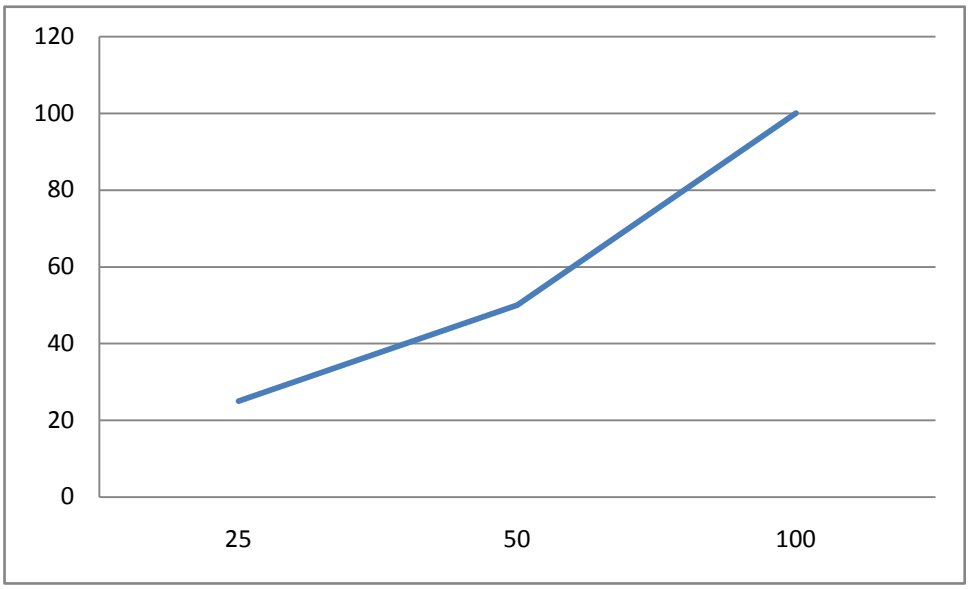

Figure 1 - ½ X Ratios

In the -½X+100 ratio the pattern was the same except reversed. For example as can be seen in Figure 2, the defects found would be 100, 50, and 25, and subsequent data

points would be incremented in a 2X fashion similar to the ½ X ratio. So the second data point was 200, 100, 50, and the third was 300, 150, and 75 respectively. This pattern was repeated through the ten data points.

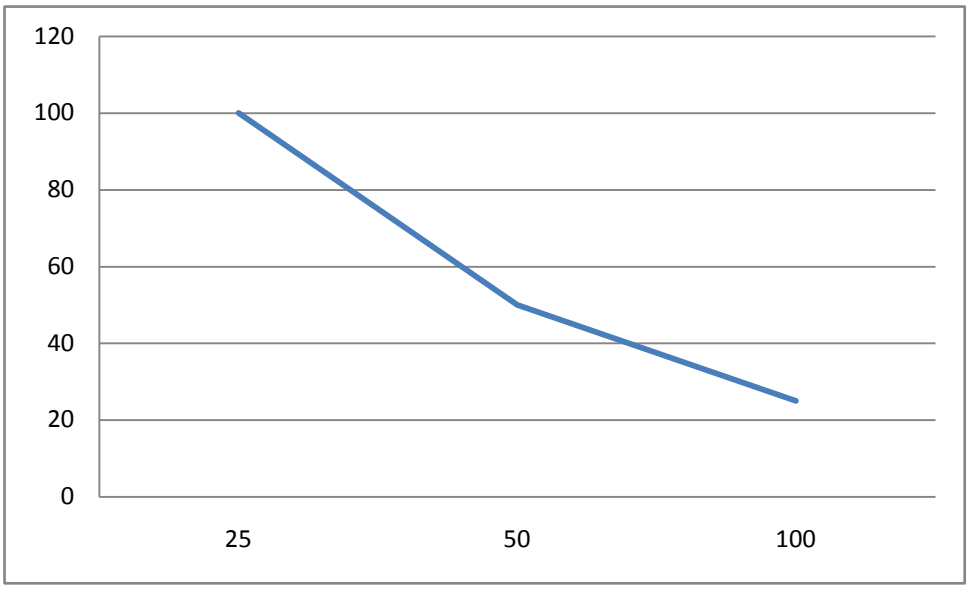

Figure 2 - (-1/2X + 100) Ratio

Chapter 6 – Results

The results of the different experiments are presented below, with the different values represented by experiment. The numbers are then graphed by data point. Each data point represents a snapshot of the experiment taken at a specific point. The different data points are listed as the first, second, third, etc, and these points represent 100 or 10%, 200 or 20%, 300 or 30%, etc. defects found out of 1000 respectively. At each data point the cost per phase, total cost, hours per phase, and total hours were captured. The numbers were then analyzed using the graphs below.

6.1 – 40 Review Hours – Linear detection – 1000 defects

The first theoretical analysis was conducted using 40 review hours with a linear detection rate. This means the rate of detection was constant for every review. For the first data point, 100 defects were found in each phase (requirements, design, and coding). A defect was defined as any issue discovered that required that required the developers to do rework on a project artifact. This definition could be applied across all reviews and was not strictly limited to a code correction. The next data point was determined at 200 defects found in each phase; the 3rd data point was calculated at 300 defects found, etc. This approach was taken for the entire first analysis. The cost per phase, as illustrated in Figure 3, increases rapidly with the requirements phase showing the least increase and the coding phase increasing the most. This is due to the higher defect correction cost associated with the later phases. The initial cost to fix all defects without reviews was $6,000,000.00. This number is constant throughout all the analyses. The cost to conduct a 40 hour review is $84,000.00; however this cost is quickly recovered, as illustrated in

Figure 4. The first data point of 100 defects found shows an initial savings of $341,000. This quick recovery of expenses results in an investment of only 120 hours. As the analysis continues the savings continue to grow $766,000 at 200 defects and $1,191,000 saved at 300 defects. These results increase in a linear manner up through 1000 defects. While the likelihood of finding 100% of the defects is extremely remote, this theoretical analysis does show that the benefits of performing reviews. As a point of reference, Capers Jones reports "As of 2007, the average for defect removal efficiency in the U.S. was about 85 percent" (2008, p. 2). At approximately 72.5% of defects found the savings actually bypass the cost. This is represented in Figure 4 where the two converging lines actually cross. If a project manager could find more than 72.5% of defects then the actual cost of the project would be outweighed by the savings and actually cut the projected cost of the project by more than half.

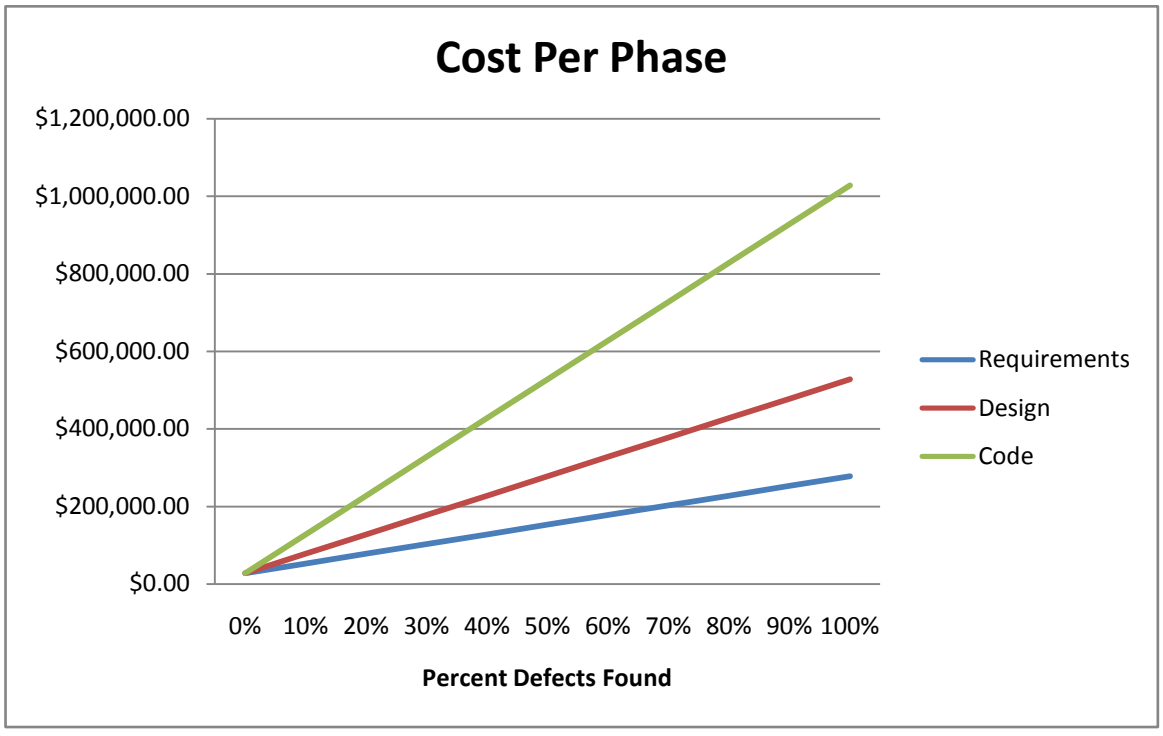

Figure 3 - Cost per phase – 40 Hour Review

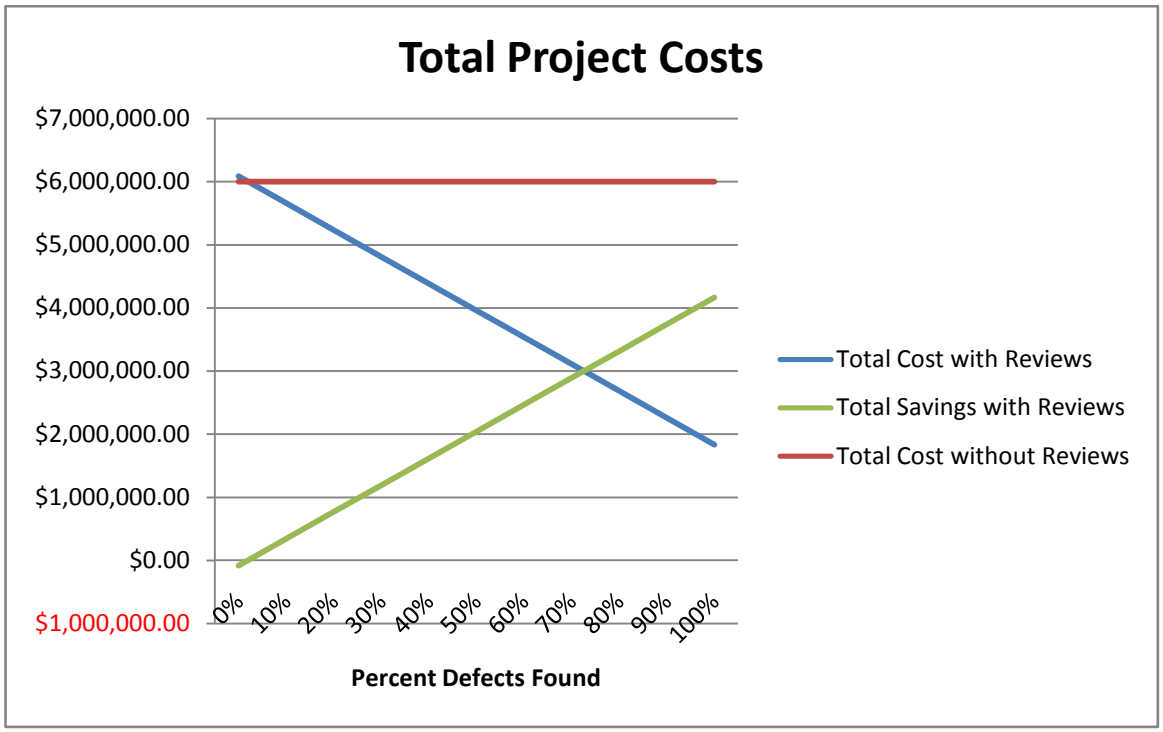

Figure 4 - Total Project Cost - 40 Hour Review

Another significant factor is the hours to complete this project. Each phase associated hours is illustrated in Figure 5. The hours for each phase increases as the amount of time to repair detected defects increases. Experience has shown that the earlier a defect is found the less time and effort is required to repair the defect, so a defect found in the requirements phase would be easier to fix than a defect found in the coding phase. Figure 5 illustrates this relationship; the requirements phase takes considerably less time to repair the 1000 defects than the coding phase's 1000 defects. The total expenditure to complete the review was 560 hours per phase for a 40 hour review. This totals to 1680 hours. The project takes a total of 144,000 hours to complete without any reviews, so the total hours to complete the project with reviews, without any cost savings from the reviews is 145,680 hours. The savings associated with the project are not just in dollars but also in time. As the data point goes up, the amount of hours to complete the

project are reduced in comparison to the total without reviews. This allows the developers to move onto different projects quicker and generate more revenue. The amount of time associated with each phase increases as the amount of defects found increase, but the overall time decreases as more defects are found. In looking at Figure 6, the initial time and effort associated with the reviews is quickly rewarded as the percentage of defects found increases.

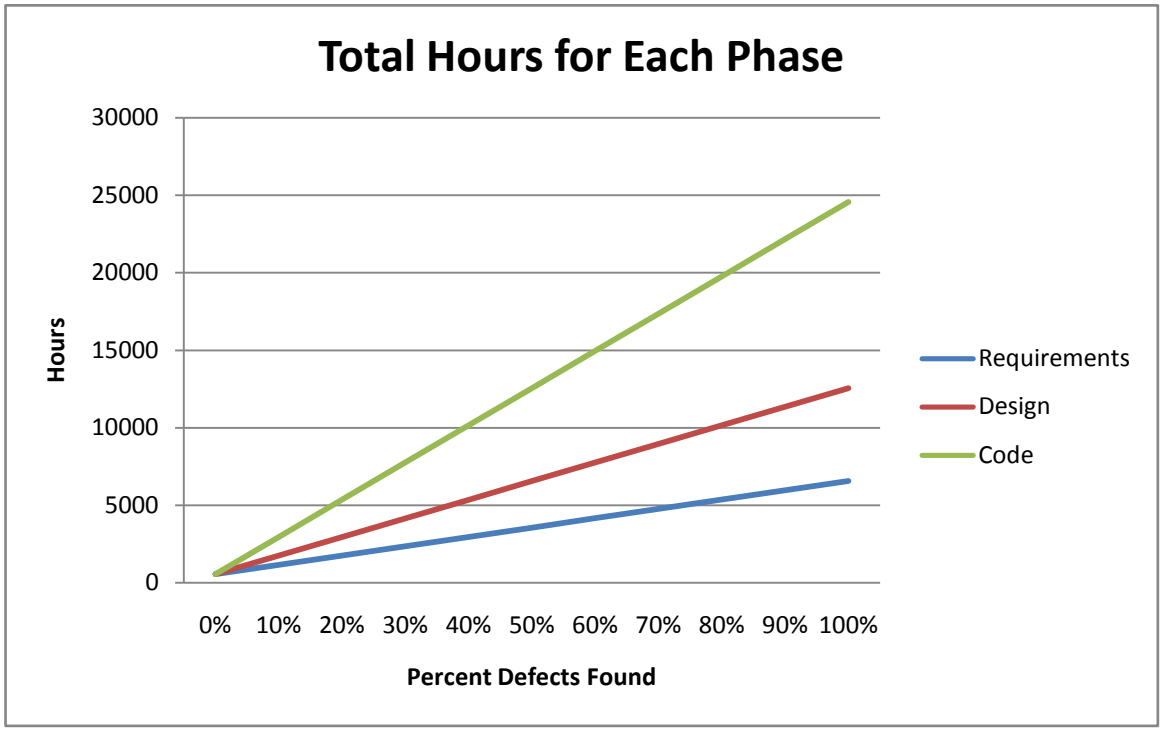

Figure 5 - Total Hours for Each Phase - 40 Hour Review

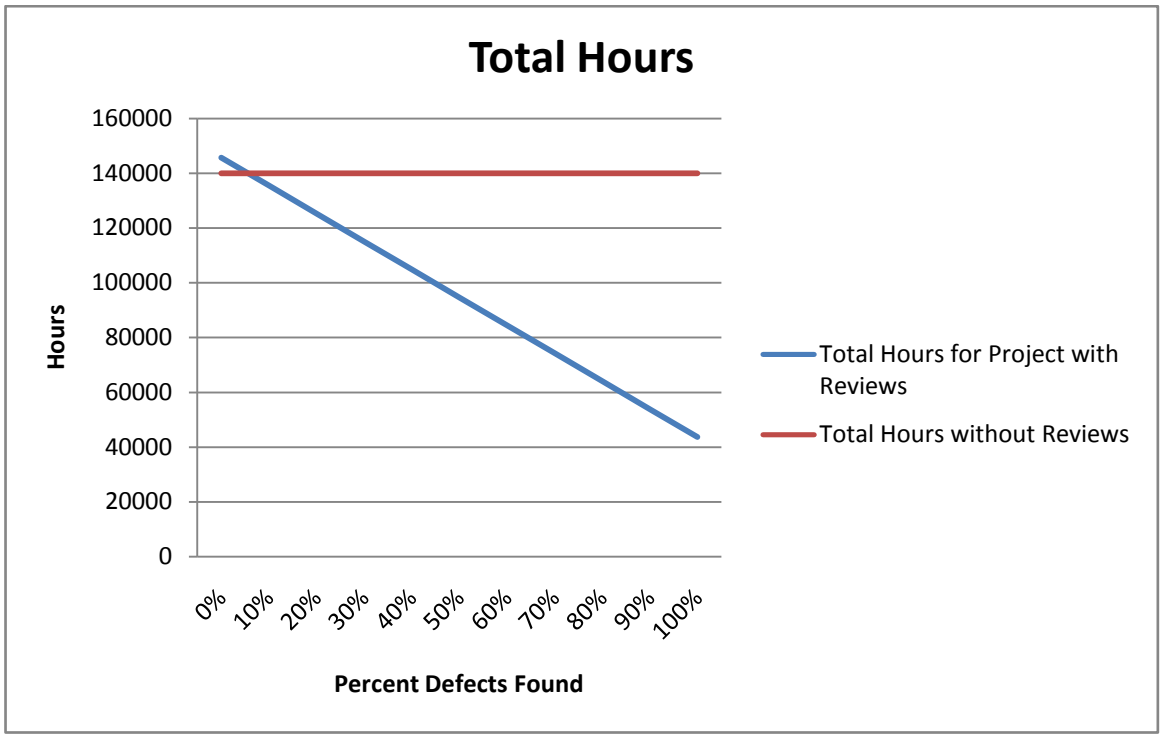

Figure 6 - Total Project Hours - 40 Hour Review

Overall, the first 40 hour review shows that, in a theoretical model, the amount of money spent on reviews is worth the time and effort. This realization comes in both time and money allocated to the project.

6.2 – 80 Review Hours – Linear detection – 1000 defect

The second theoretical analysis was an 80 hour review; like the 40 hour review the rate of defect detection was linear with the first data point at 100 defects, the second at 200 defects, etc. As one can see in Figure 7, the cost per phase remains linear which is consistent with the previous analysis. The initial cost of the reviews was $168,000 dollars. Again the total cost of defect correction without reviews was $6,000,000. This cost was quickly recovered within the first data point. The total savings for the first data

point was $257,000. The cost savings was linear in nature as shown in Figure 8, and the cost savings realized surpass the total cost of the project.

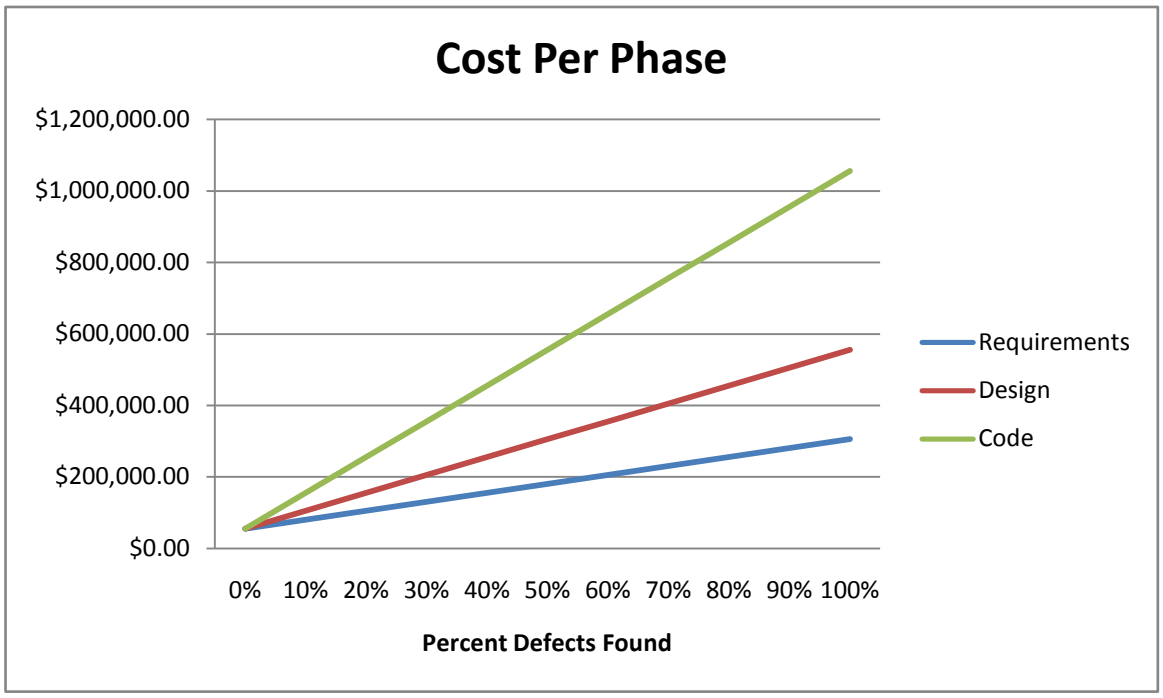

Figure **7 - Cost per Phase - 80 Hour Review**

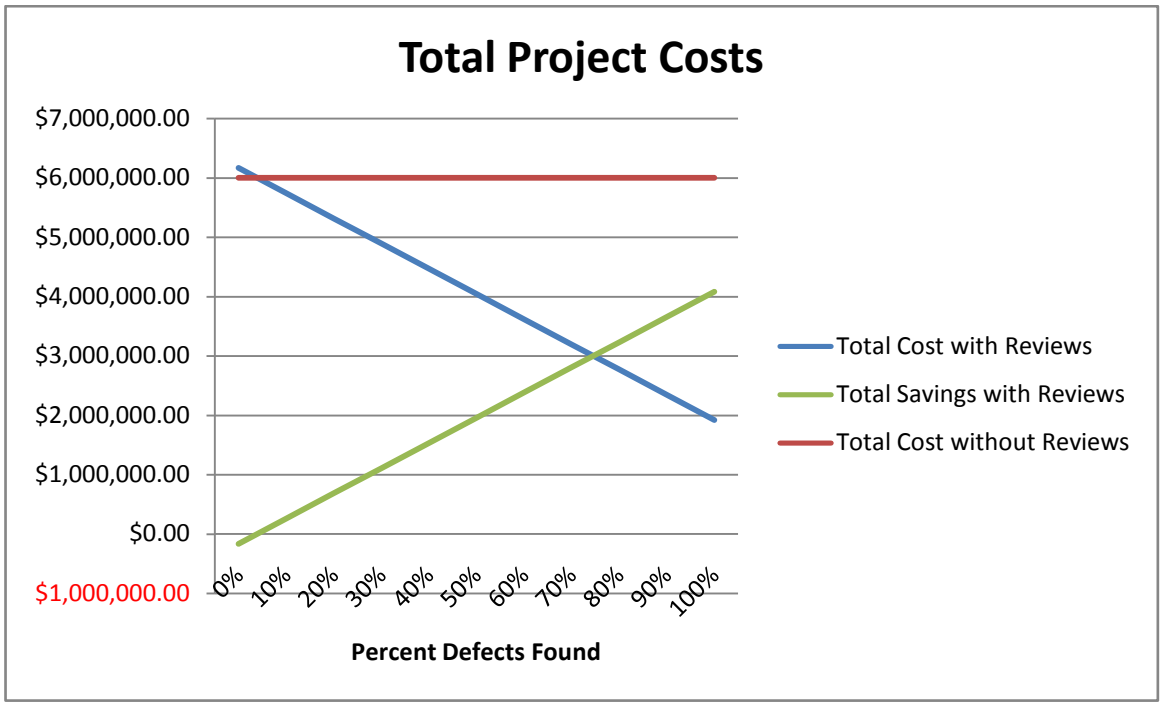

Figure 8 - Total Project Cost - 80 Hour Review

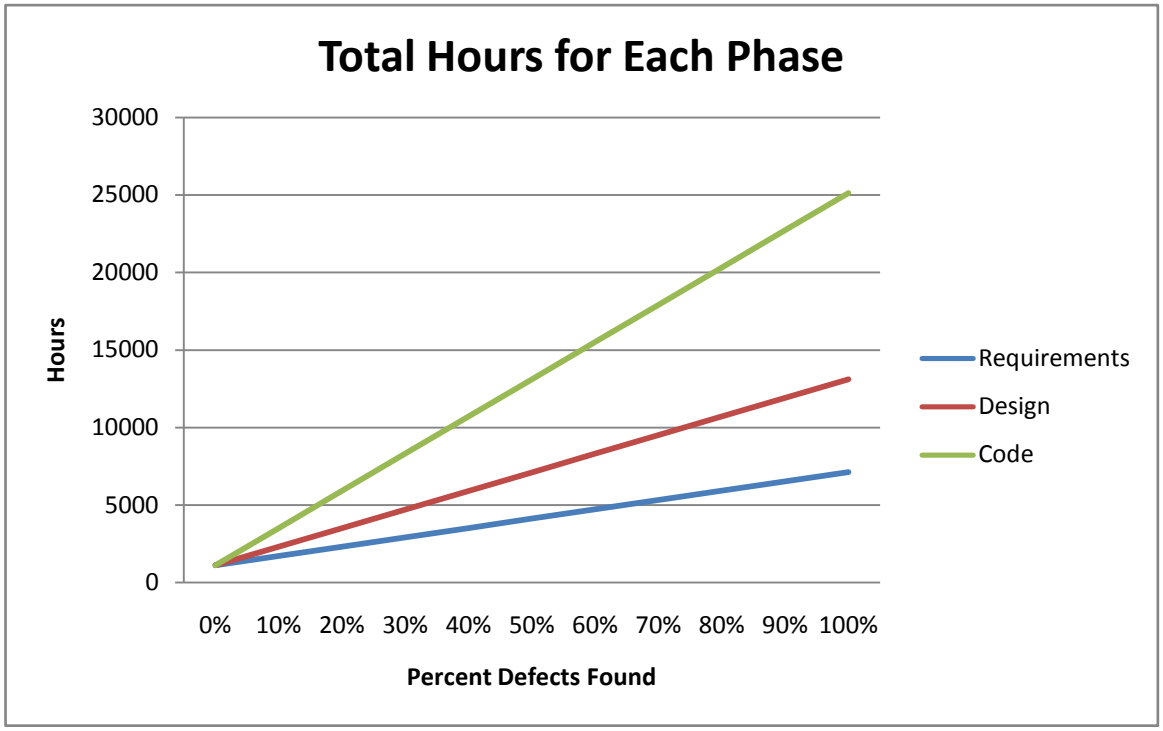

Figure 9 - Total Cost per Phase - 80 Hour Review

The total hours for the project was 144,000 hours, however with 80 hour reviews built in that number grows to 147,360 hours. The actual total hours per phase remain linear, and as can be seen in Figure 9, they increase as the review hours increase. These totals are assuming there are no defects found. When one analyzes the data to 50% of the defects found in Figure 10, a reviewer will notice the total hours to complete the entire project shrinks to 96,360 hours. This number continues to fall as the data point moves to towards 100%.

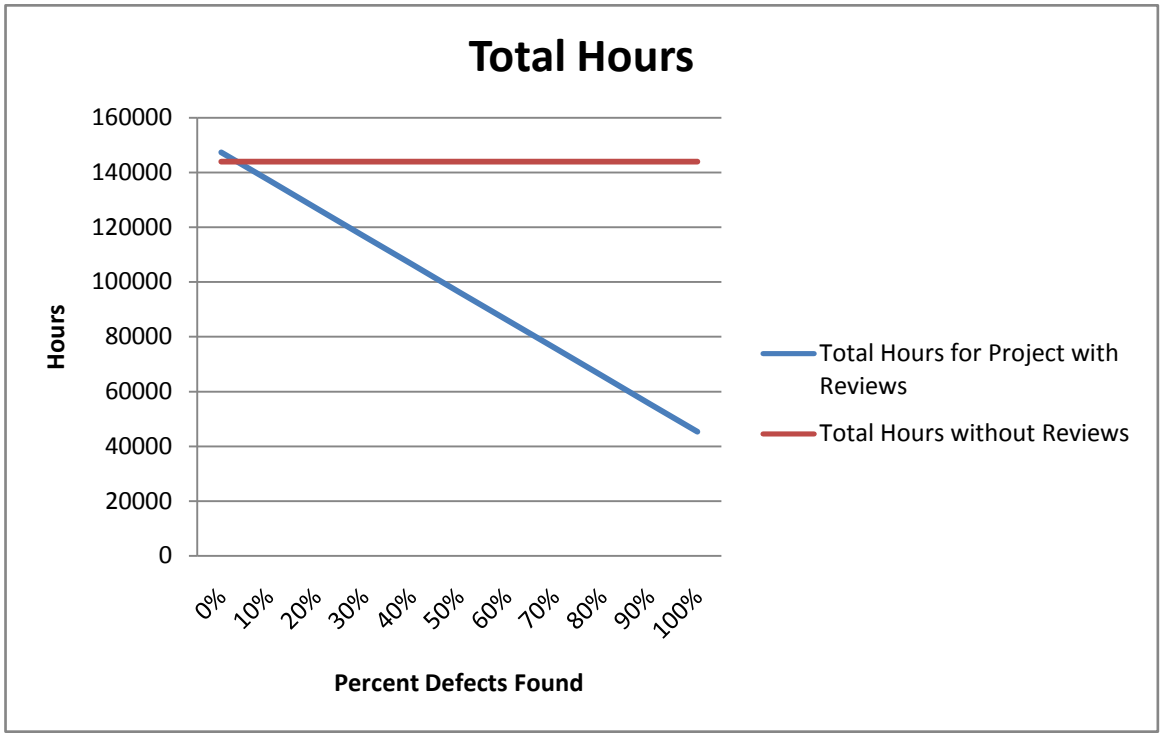

Figure 10 - Total Hours - 80 Hour Review

6.3 – 160 Review Hours – Linear detection – 1000 defect

The final linear detection theoretical analysis was the 160 hour review. This review began with the $6,000,000 beginning cost and when the 160 hour reviews were added, the cost went to $6,336,000, an increase of $336,000 dollars. Figures 11 and 12 show that as with the other linear detection analyses, the overall cost of the reviews was

recouped before the first data point of 100 defects. In this case the savings was $89,000, and increased to $1,789,000 at the fifth data point of 500 or ½ of the defects found. This is a significant yet attainable goal when one considers the research done by Capers Jones in 2008 showed that the average was 85% (p. 2). The 72.5% point again is where the saving again actually passes the cost of the project.

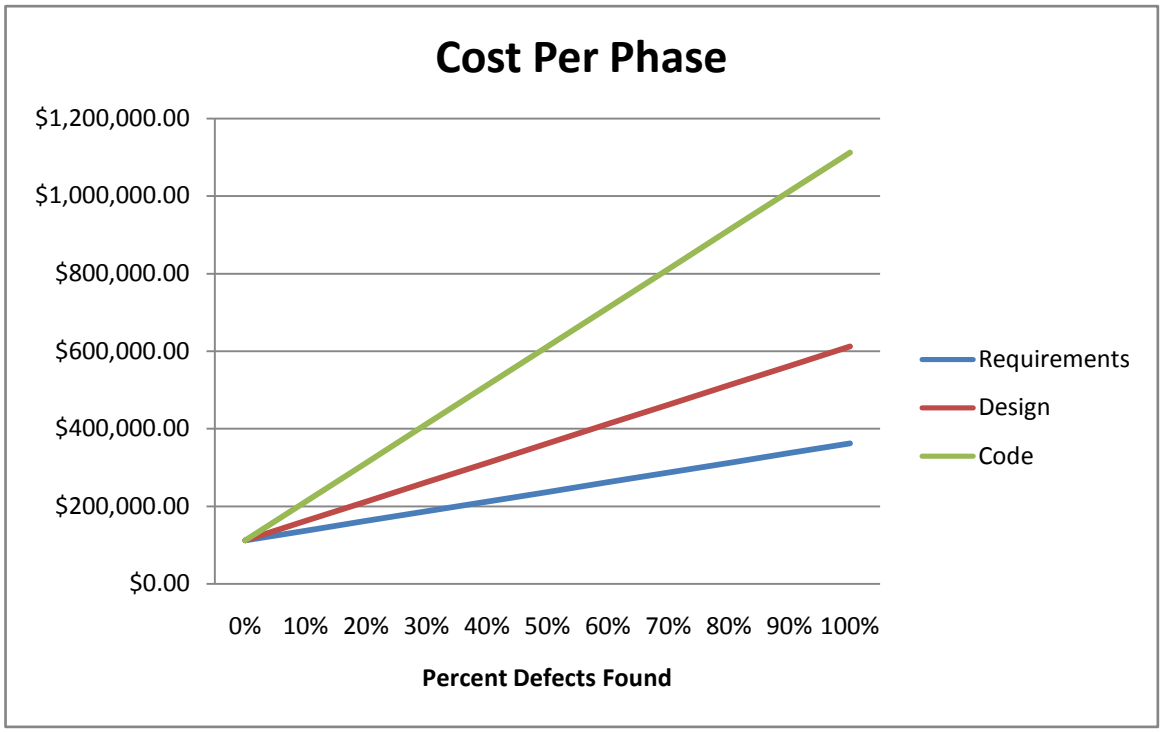

Figure 11 - Cost per Phase - 160 Review Hours

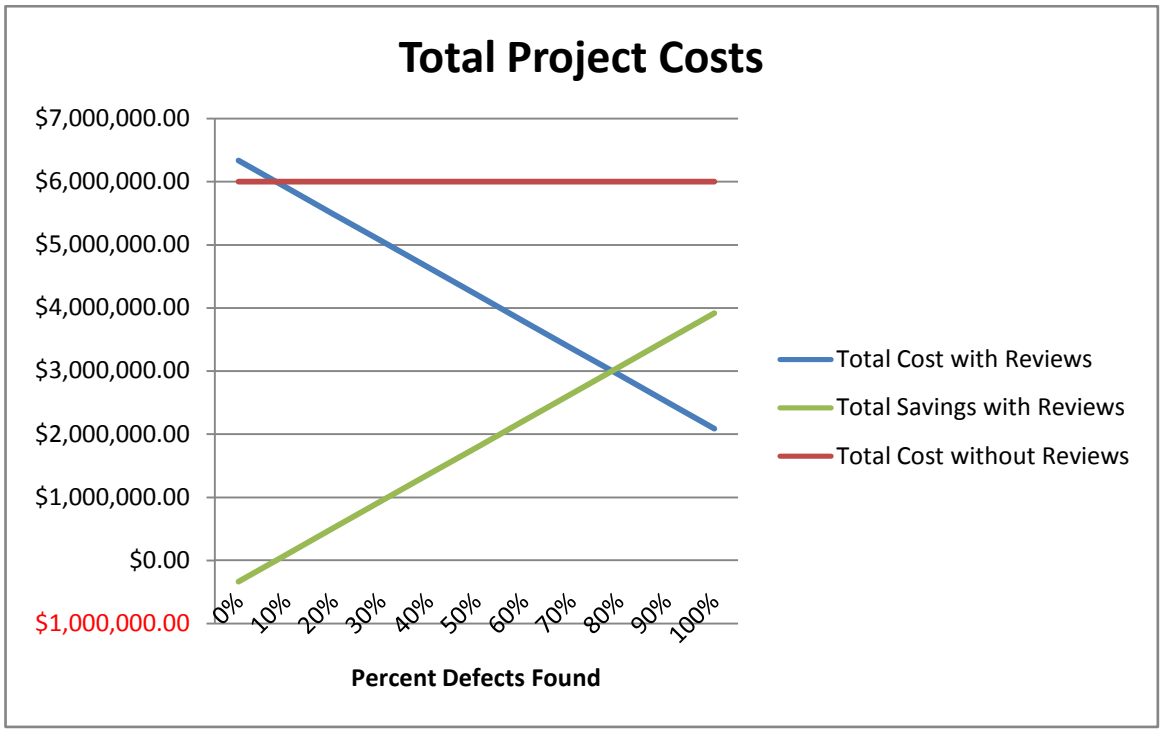

Figure 12 - Total Project Cost - 160 Review Hours

When considering Figures 13 and 14, it is easy to see that the most significant savings is again in time. The overall hours for the project are 144,000, and when the reviews are figured in that number grows to 150,720 hours. When the fifth data point is reached the number of total hours has shrunk to 99,720 hours.

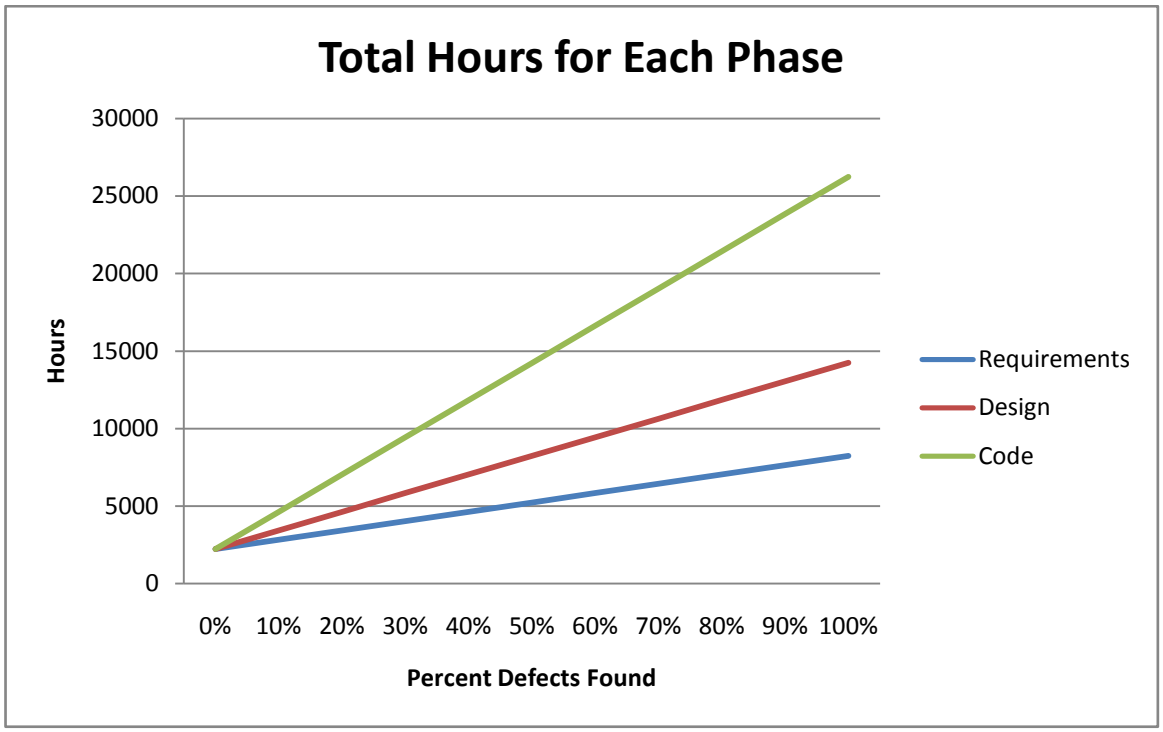

Figure 13 - Total Hours per Phase - 160 Review Hours

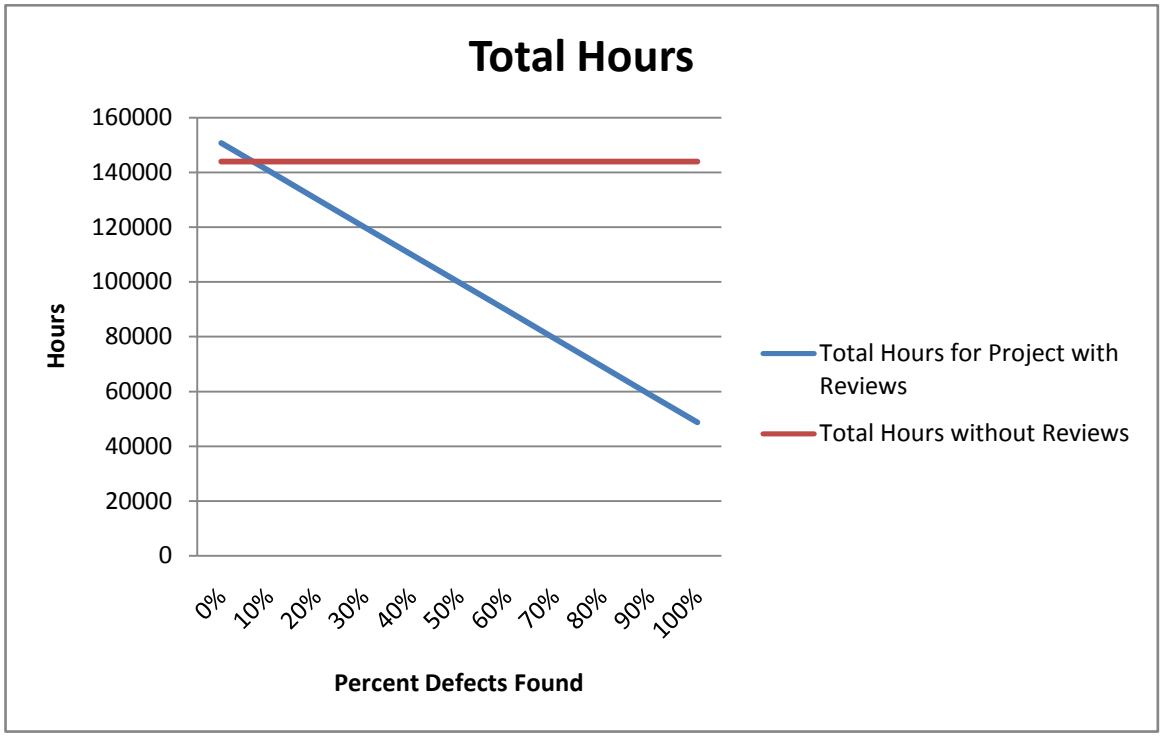

Figure 14 - Total Hours - 160 Review Hours

Overall the linear detection rate model presents an interesting theoretical review of how the use of reviews would be beneficial; however this model is not very helpful or realistic. A company that spends considerable time on reviews and finds defects in this linear fashion would find it nearly impossible to follow these quantitative relationships. The ability to find defects in design and requirements is certainly easier than in the coding phase where test data is required and test driver software is typically written, therefore a better model would be to increase the amount of defects found in the requirements phase and then decrease the amount incrementally thru the other two phases.

6.4 – 40 Review Hours - $-\frac{1}{2}X + 100$ Detection Rate - 1000 defect

The next several theoretical analyses were conducted using a $-\frac{1}{2}X +$ 100 detection rate. As was explained earlier, this detection rate consisted of finding 100 defects for the 1st or requirements phase, 50 for the design phase, and 25 for the coding phase. The number of defects detected were increased by 2X for each data point, so for the second data point, the number of defects detected were, 200, 100, and 50, and for the third data point the number of defects detected was 300, 150, and 75. This pattern was continued until the requirements phase defects detected reached 1000. The final defect detection numbers were 1000, 500, and 250. Figure 15 shows that the total cost per phase are the same in this theoretical analysis; this is due to the reduction in the number of defects found, and therefore the number of hours to repair them. As can be seen in Figure 16, the initial cost for conducting the reviews was $84,000, and the initial cost to repair all defects without reviews remained at $6,000,000, therefore the total cost with

reviews was $6,840,000. As with the other analyses the cost of the review was quickly recovered. This is consistent with the other theoretical analyses conducted with a linear detection rate; however the time it took to recover the costs of the reviews was pushed to the right.

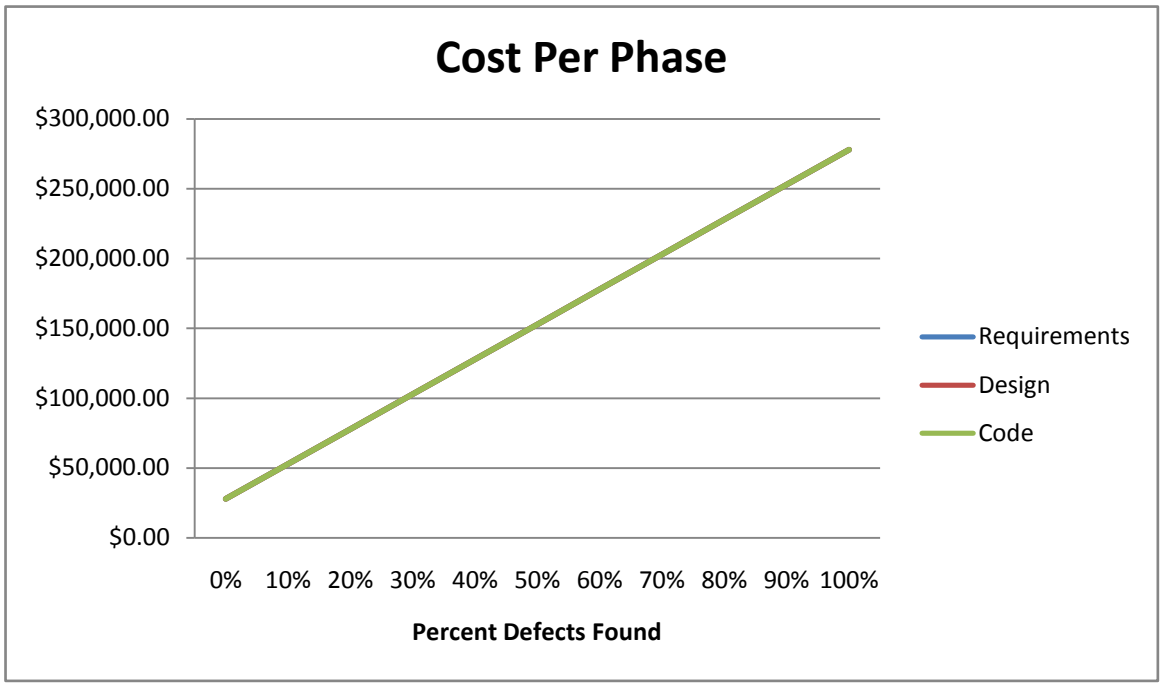

Figure 15 - Cost Per Phase - $-\frac{1}{2}X + 100$ Detection Rate - 40 Hour Review

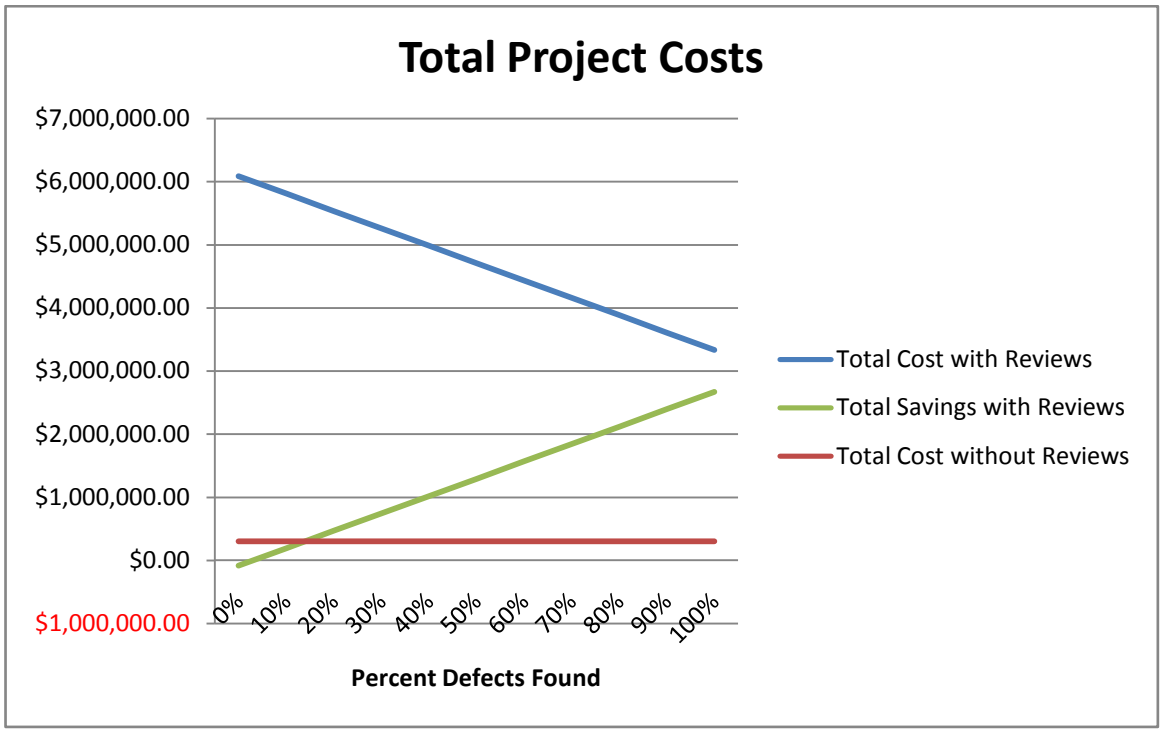

Figure 16 - Total Project Costs - $-\frac{1}{2}X + 100$ Detection Rate - 40 Hour Review

It is interesting to note, that Figure 16 clearly shows that the cost of the project never falls below the cost savings of the reviews. This theoretical analysis is more indicative of a real project; however the model is still theoretical in nature. This increased cost is due to the failure to find all the defects during the three phases. There are a significant number of defects remaining in the project that are found and repaired using the cost associated with the development phase.

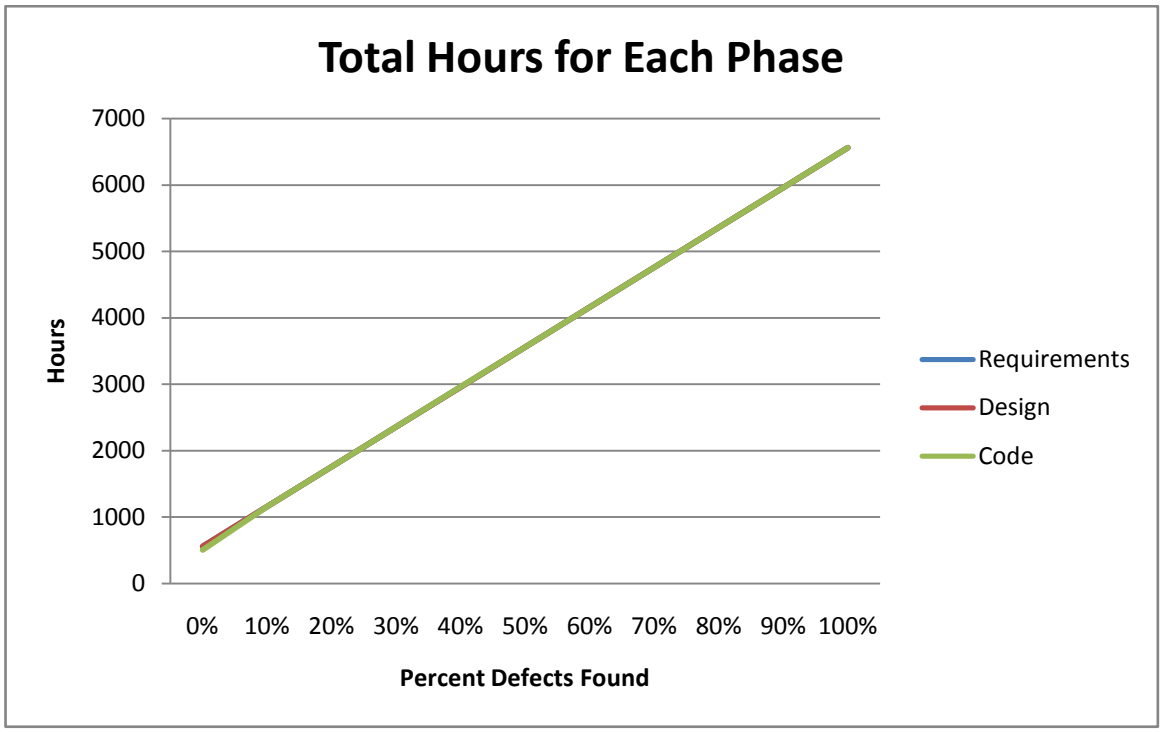

Figure 17 - Total Hours per Phase - $-\frac{1}{2}X + 100$ Detection Rate - 40 Hour Review

Due to the nature of the number of hours needed to repair a defect, and the negative nature of the number of defects found, the number of hours required for each phase was the same, this is illustrated in Figure 17, and as can be seen in Figure 18, the total hours for the project declined at a steady rate, however it was less than the linear detection rate.

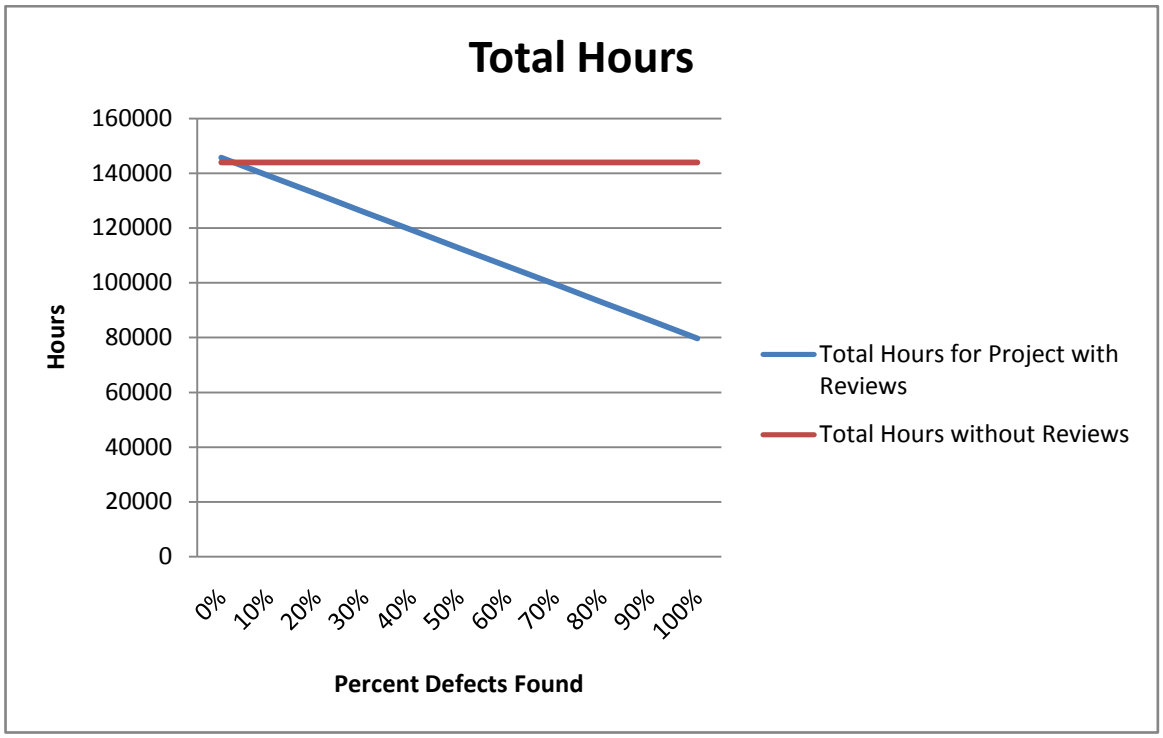

Figure 18 - Total Hours - $-\frac{1}{2}X + 100$ Detection Rate - 40 Hour Review

6.5 – 80 Review Hours - $-\frac{1}{2}X + 100$ Detection Rate - 1000 defect

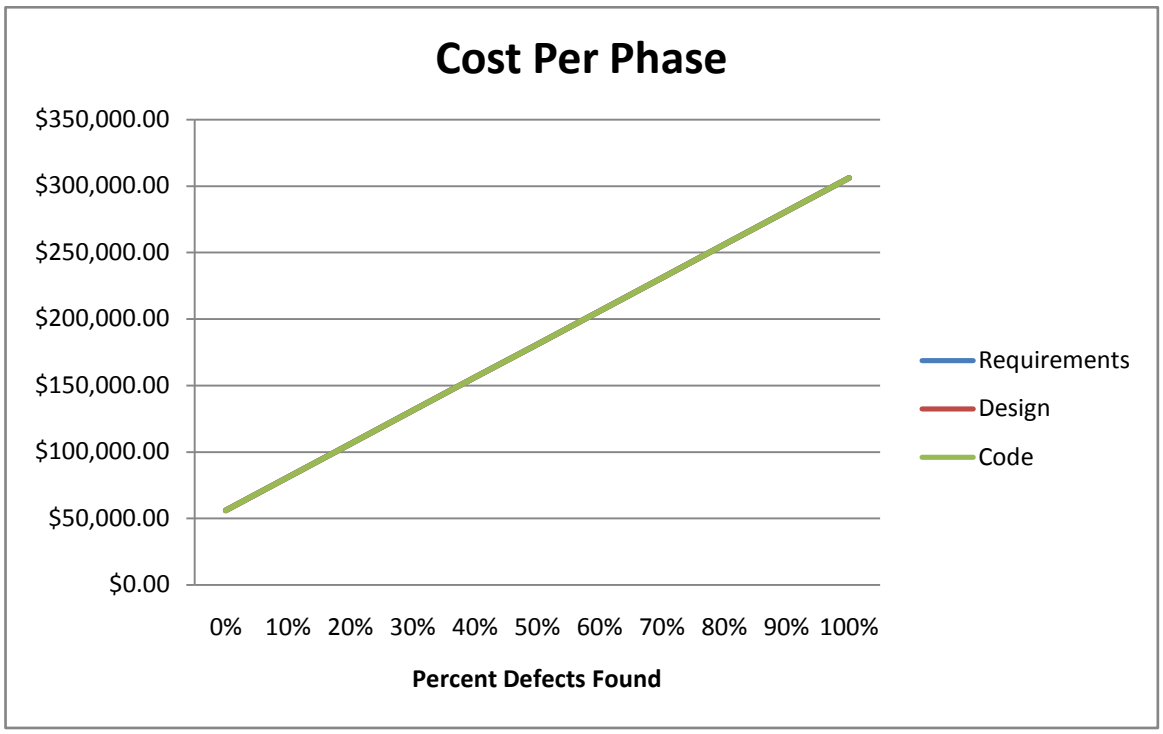

Figure 19 – Review Hours - $-\frac{1}{2}X + 100$ Detection Rate - 80 Hour Review

The next theoretical analysis continued the $-\frac{1}{2}X + 100$ detection rate study. Figure 19 illustrates, that this theoretical analysis continued to show the consistent nature of the costs per phase and Figure 20 shows the same for the total project costs. The overall costs were initially increased by $168,000 but again that was quickly recovered in the first data point. The total project costs were calculated to be $3,418,000 with this detection rate assuming 100% of the defects were found.

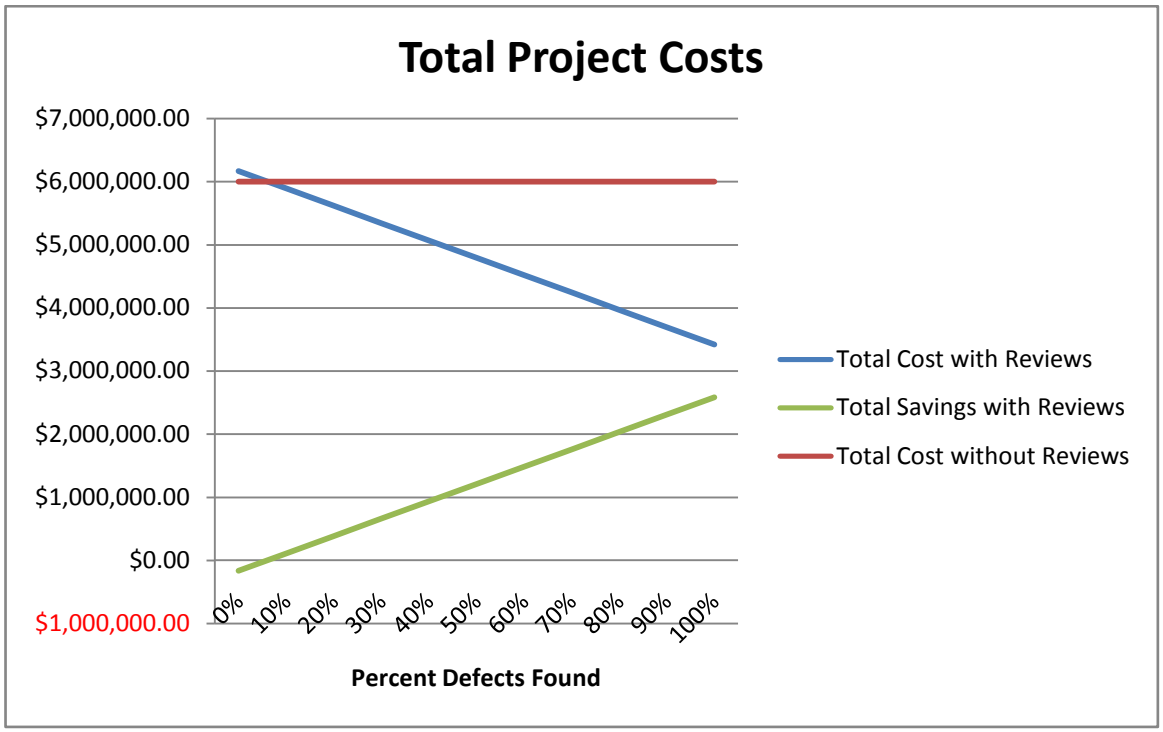

Figure 20 - Total Project Cost - $-\frac{1}{2}X + 100$ Detection Rate - 80 Hour Review

The hours per phase remained constant between the phases as seen in Figures 21 and Figure 22 shows the total hours for the project were calculated to be 147,360. This is in comparison to 144,000 with no reviews; however the overall total project hours were calculated to be 140,760 at the first data point, 114,360 hours if 50% at the 5th data point and finally 81,360 hours at the 10th and final data point. This presents a significant person hour savings and could be extrapolated to a real calendar time savings, thereby allowing a product to get to market faster.

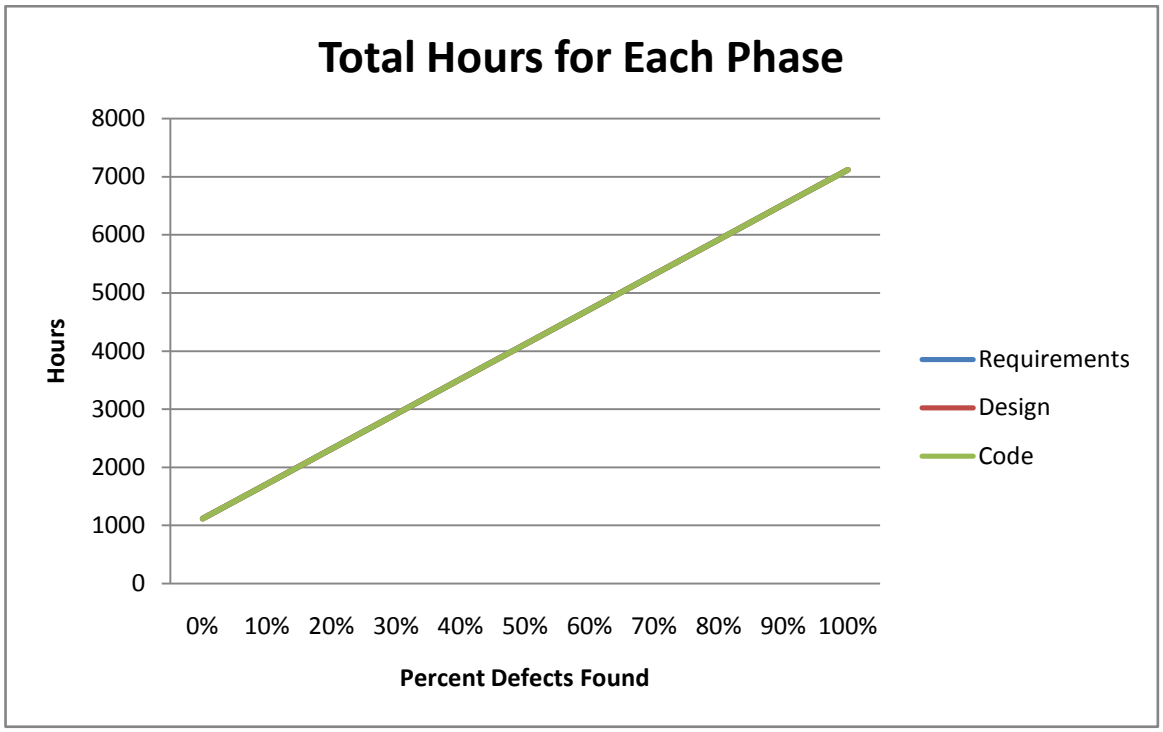

Figure 21 - Total Hours per Phase - $-\frac{1}{2}X + 100$ Detection Rate - 80 Hour Review

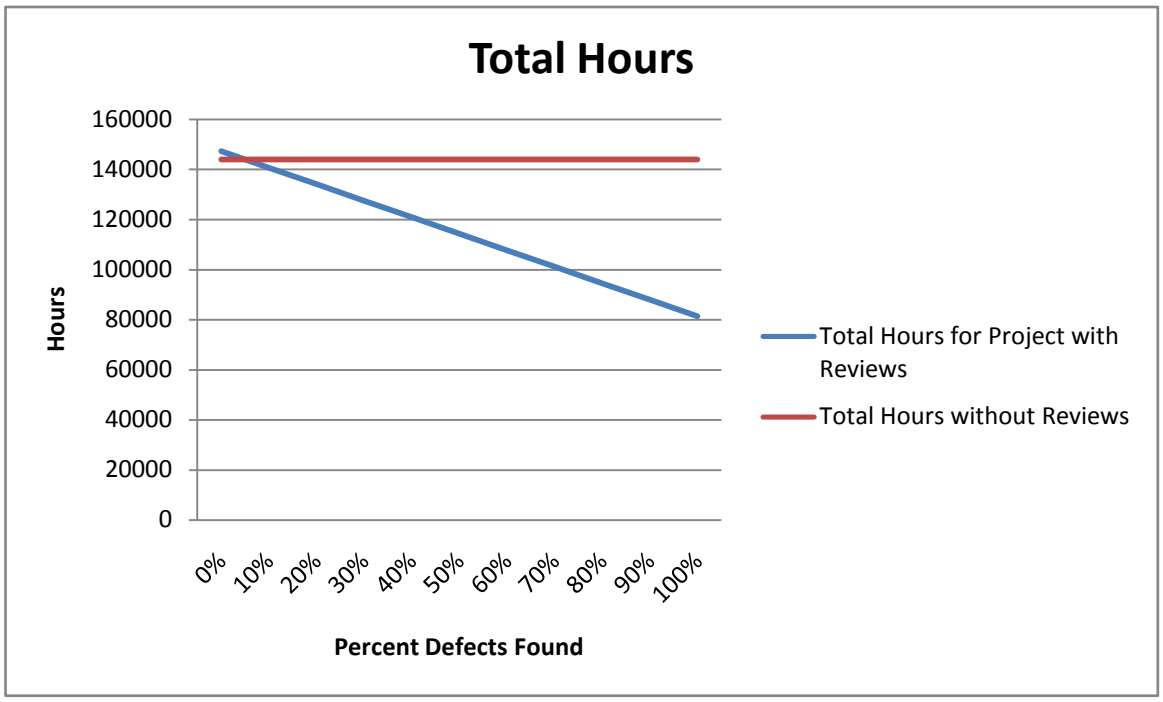

Figure 22 - Total Hours - $-\frac{1}{2}X + 100$ Detection Rate - 80 Hour Review

6.6 – 160 hour Review Hours - $-\frac{1}{2}X + 100$ Detection Rate - 1000 defect

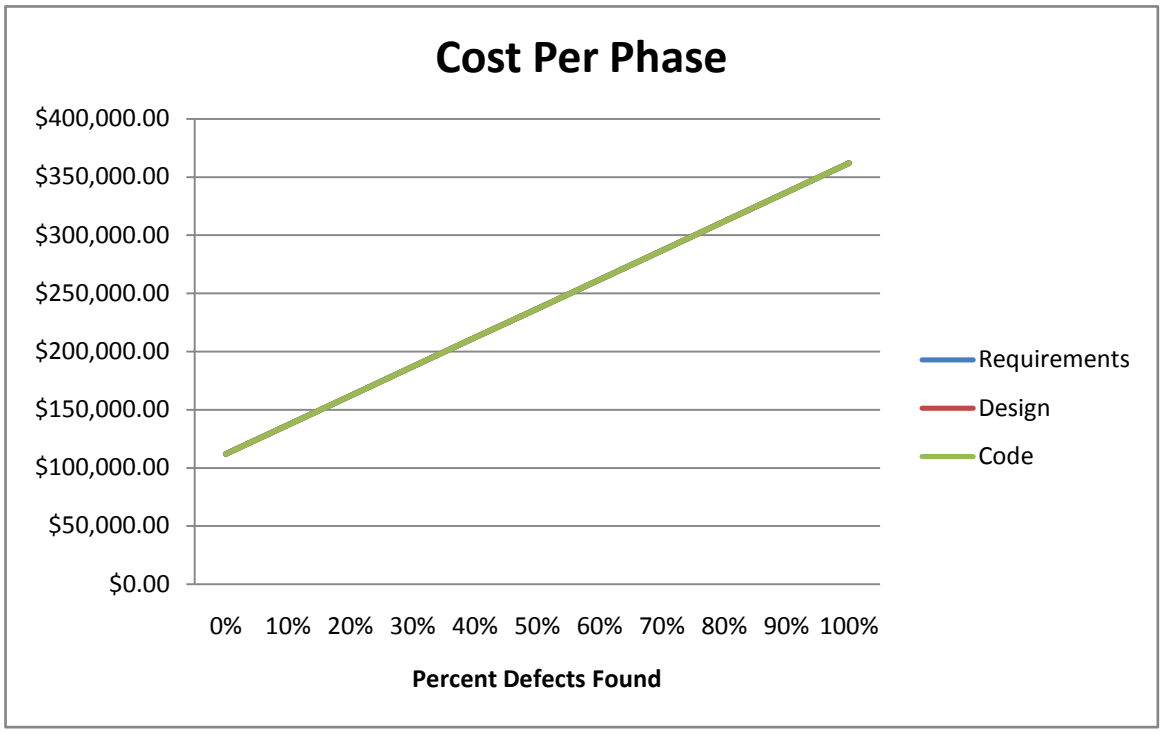

Figure 23 - Total Hours - $-\frac{1}{2}X + 100$ Detection Rate - 160 Hour Review

The next theoretical analysis was conducted using 160 review hours. Figures 23 and 24 illustrate this theoretical analysis again have the same trends as the other $-\frac{1}{2}X + 100$ detection rate analyses. This theoretical analysis had an overall cost of $6,336,000 with the 160 hour review; however the difference between this theoretical analysis and the others is that recovering the cost of the reviews was not completed until the 2nd data point. There was still a $61,000 net loss after the 1st data point. The overall savings for this detection rate with the 160 hour review was $3,586,000.

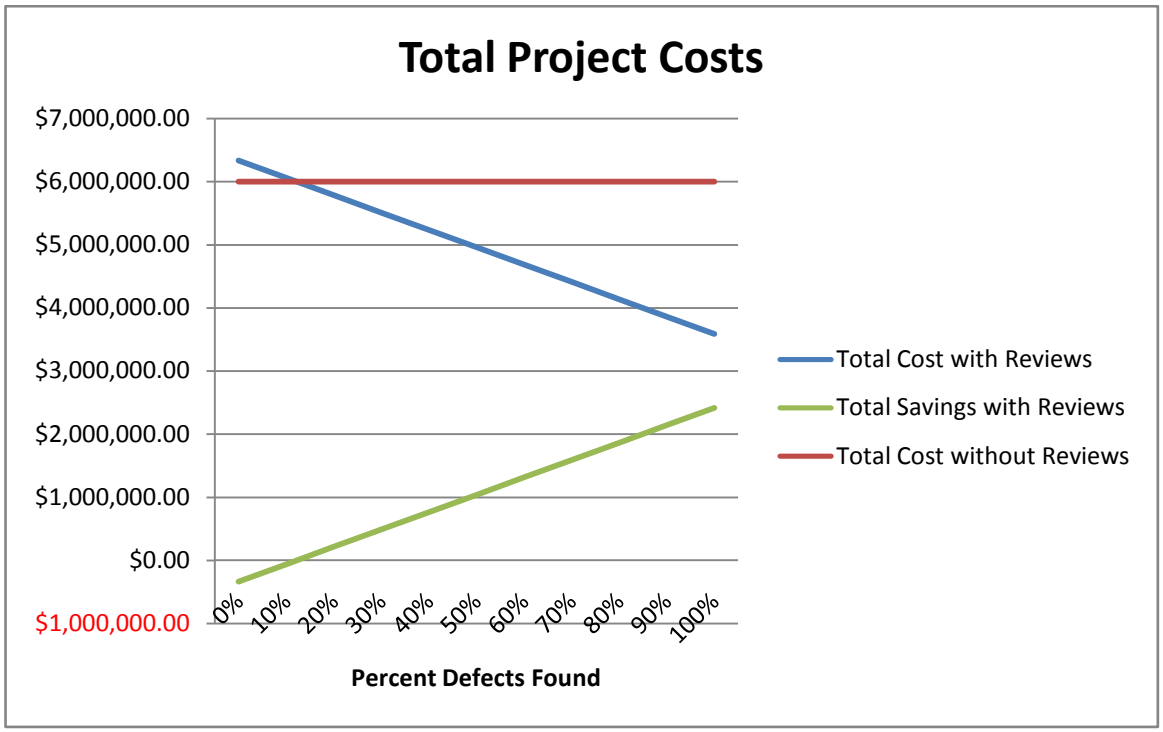

Figure 24 - Total Hours - $-\frac{1}{2}X + 100$ Detection Rate - 160 Hour Review

Figure 25 demonstrates that the total hours per phase in this analysis remains consistent with the other analyses. The total hours also show a slight difference also, as demonstrated in Figure 26, reflecting the increased amount of time dedicated to the review for a total of 150,720 hours which shows an overall increase of 6720 hours for the reviews. The first data point still shows a positive 120 hours between the project hours with reviews and the total project with no reviews; however the cost of review in hours was made up in the next data point. The total hours at the 5th data point was 117,720, and the last data point had total hours of 84,720.

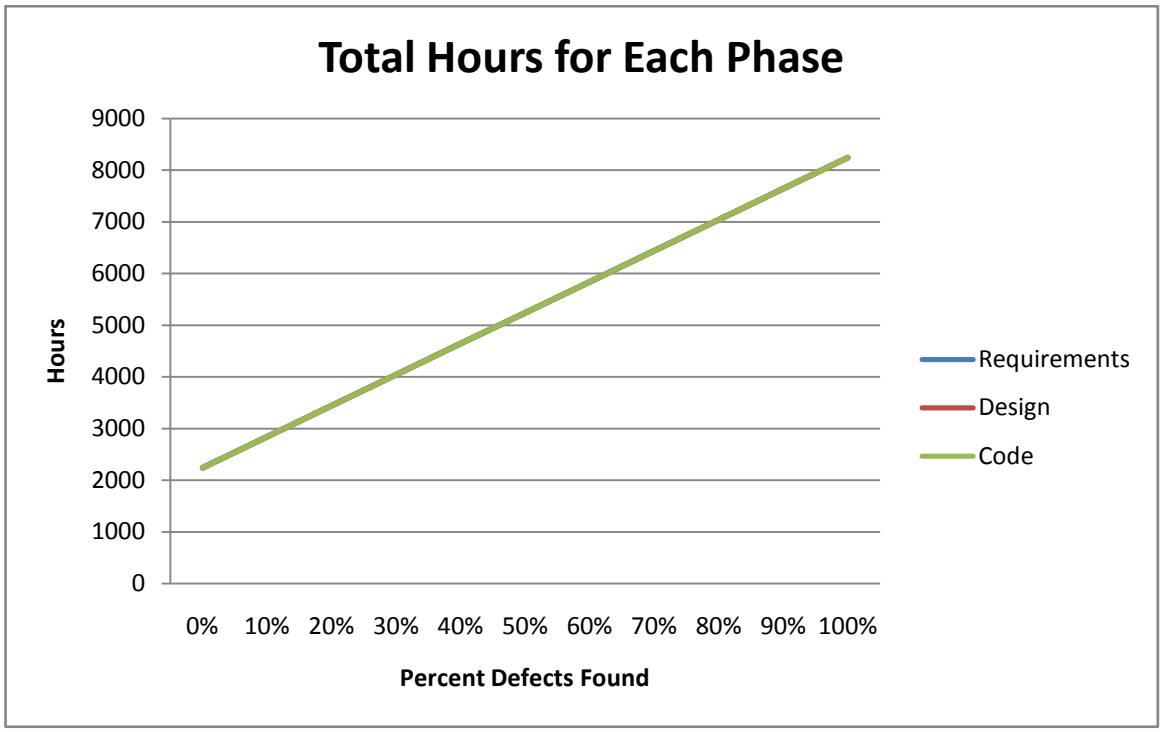

Figure 25 - Total Hours - $-\frac{1}{2}X + 100$ Detection Rate - 160 Hour Review

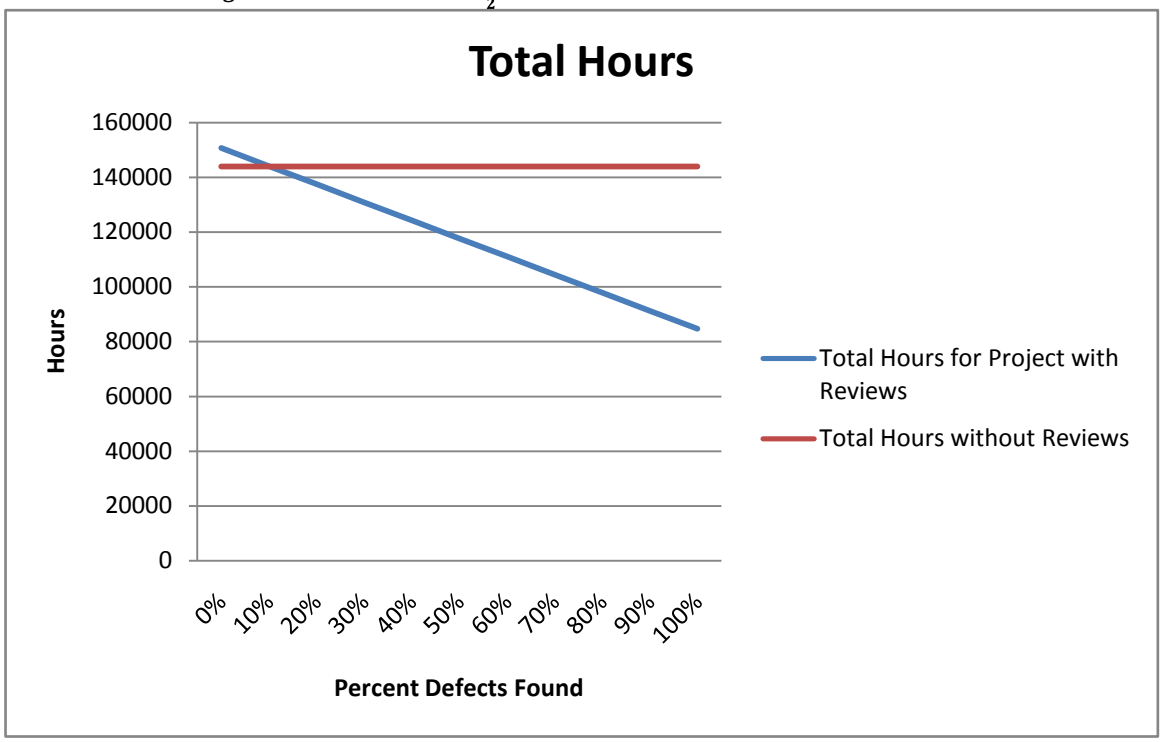

Figure 26 - Total Hours - $-\frac{1}{2}X + 100$ Detection Rate - 160 Hour Review

6.7 – 40-80-160 hour Review Hours - 1000 defect

In the next theoretical analysis the number of review hours was altered for each phase. In the first phase or the requirements phase, the review was 40 hours, in the design phase the review hours were 80 hours and in the final coding phase, the review was 160 hours. This arrangement affected the outcome significantly. Figure 27 displays the cost per phase, showing that the cost per phase has increased significantly from the other analysis. The next figure, Figure 28, shows the total cost for the project, the total savings for the project, and the cost for a project without reviews. As can be seen, the cost for the entire project passes the cost savings again at the 72.5% defect found mark. This illustrates the theoretical importance of doing reviews. The first data point showed a savings of $229,000. The 5th data point showed a savings of $1,929,000 and the final data point had a savings of $4,054,000. The actual cost of the project, if 100% of the defects were found was $1,946,000. This is in comparison to the same project with no reviews costing $6,000,000. While this is theoretically compelling, an actual project would be highly unlikely to obtain the results given factors such as requirements volatility, personnel turnover, technical and tool overhead, documentation, contractual obligations, and other real world issues.

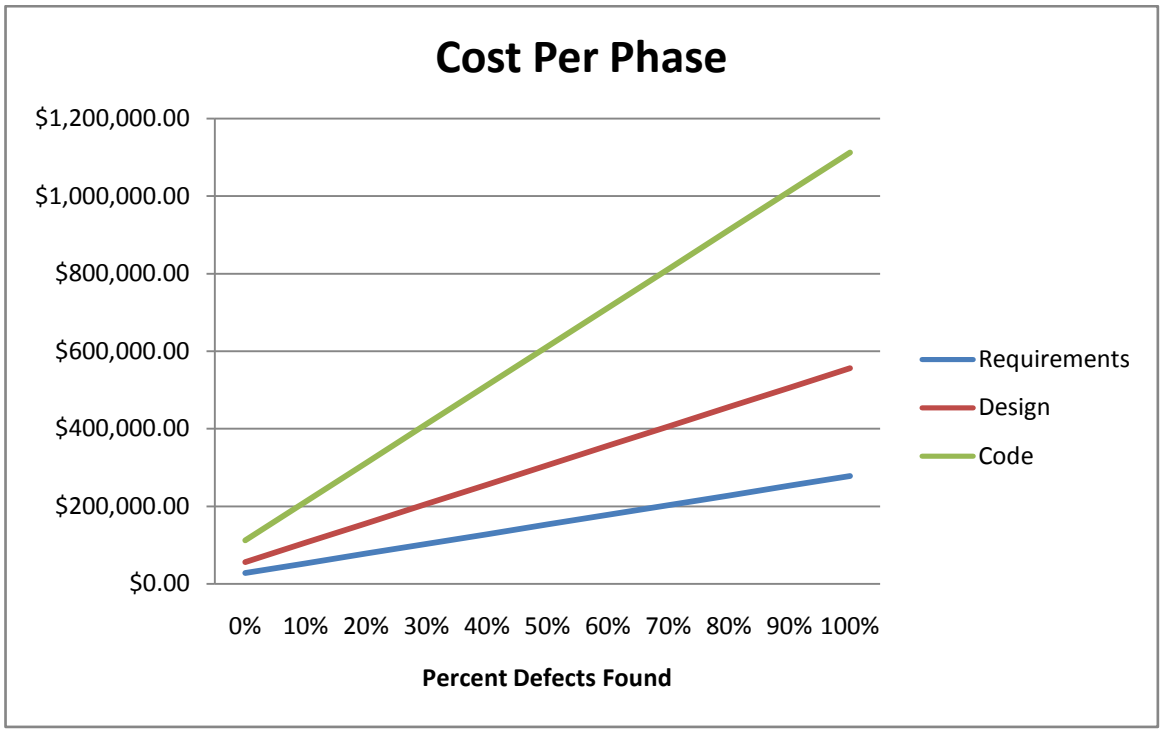

Figure 27 - Cost per Phase - 40-80-160 Hour Review

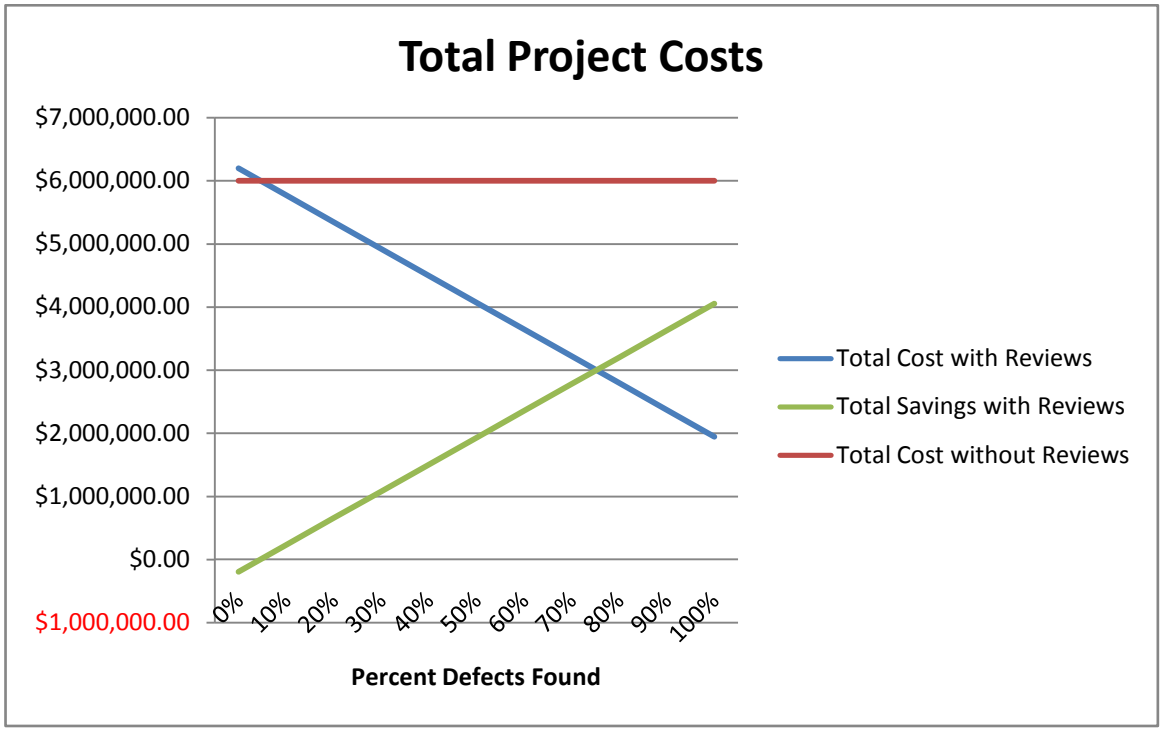

Figure 28 - Total cost - 40-80-160 hour review

The hours were also impacted significantly. Figure 29 exhibits this significant increase in hours per phase, and the 1st data point in Figure 30 shows a total of 137,720 hours, with the 5th data point having 96,920 hours, and finally in the 10th data point the total hours for the project was 45,920 hours compared to a total of 144,000 hours if no reviews were used.

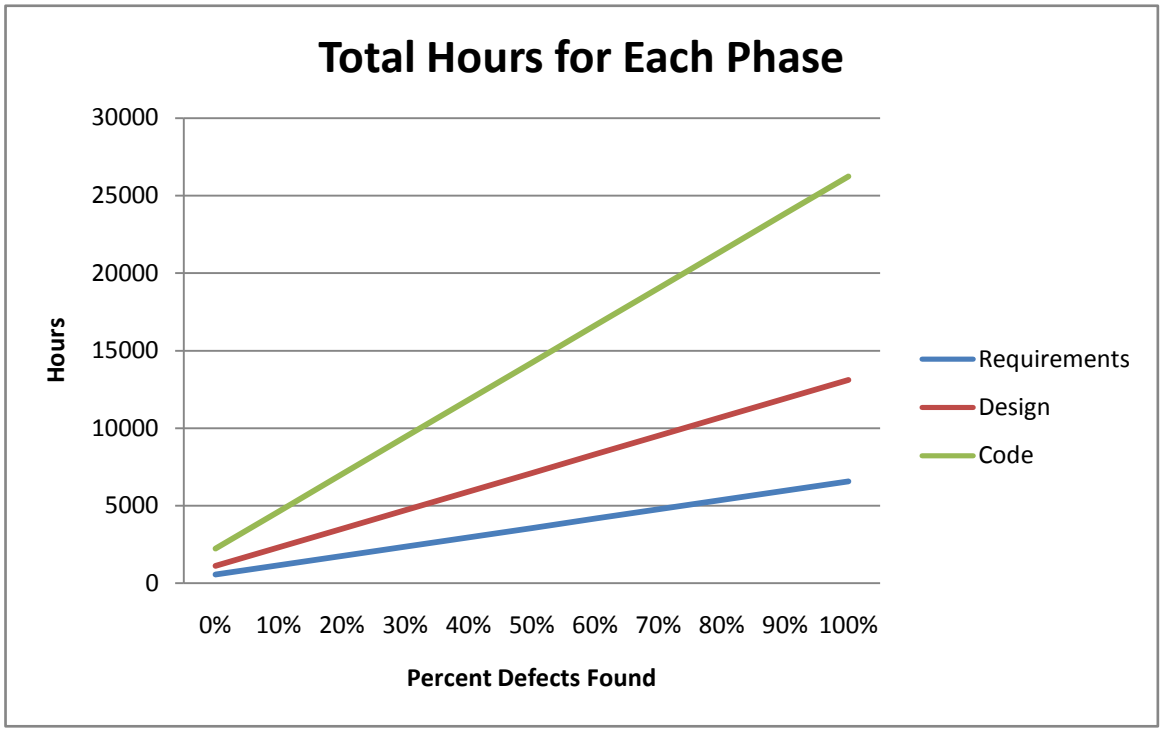

Figure 29 – Hours by Phase - 40-80-160 Hour Review

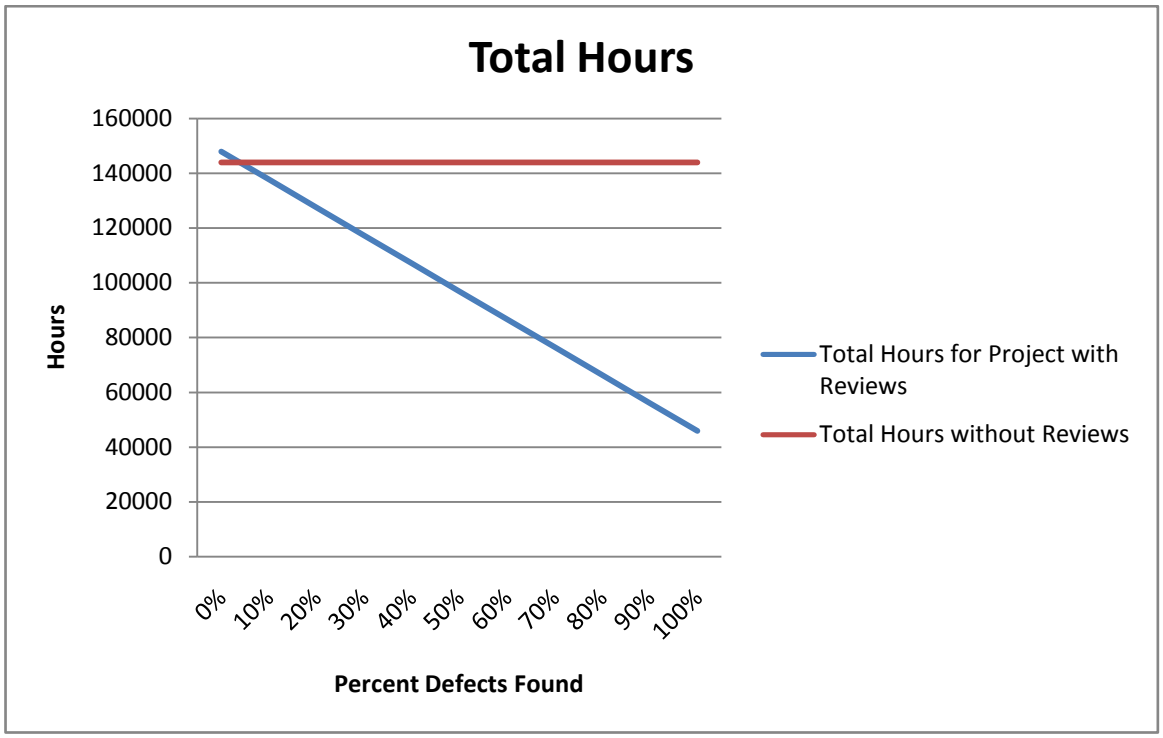

Figure 30 - Total Hour - 40-80-160 Hour Review

6.8–40-80-160 Hr Review Hours - $-\frac{1}{2}X + 100$ Detection Rate - 1000 defect

This test was conducted using 40 hours, 80 hours, 160 hours in the different phases, requirements, design, and coding respectively. The cost per phase shows a significant difference than the other analysis. Figure 31 shows the cost for each phase having a different initial cost with the difference maintained constantly as the project moves through to completion. The initial cost of the project with the reviews was $6,196,000. This initial cost was quickly recouped, again within the first data point. When the 5th data point was reached the total savings was $1,179,000 dollars, and when the 10th data point was reached the savings was $2,554,000 as illustrated in Figure 32.

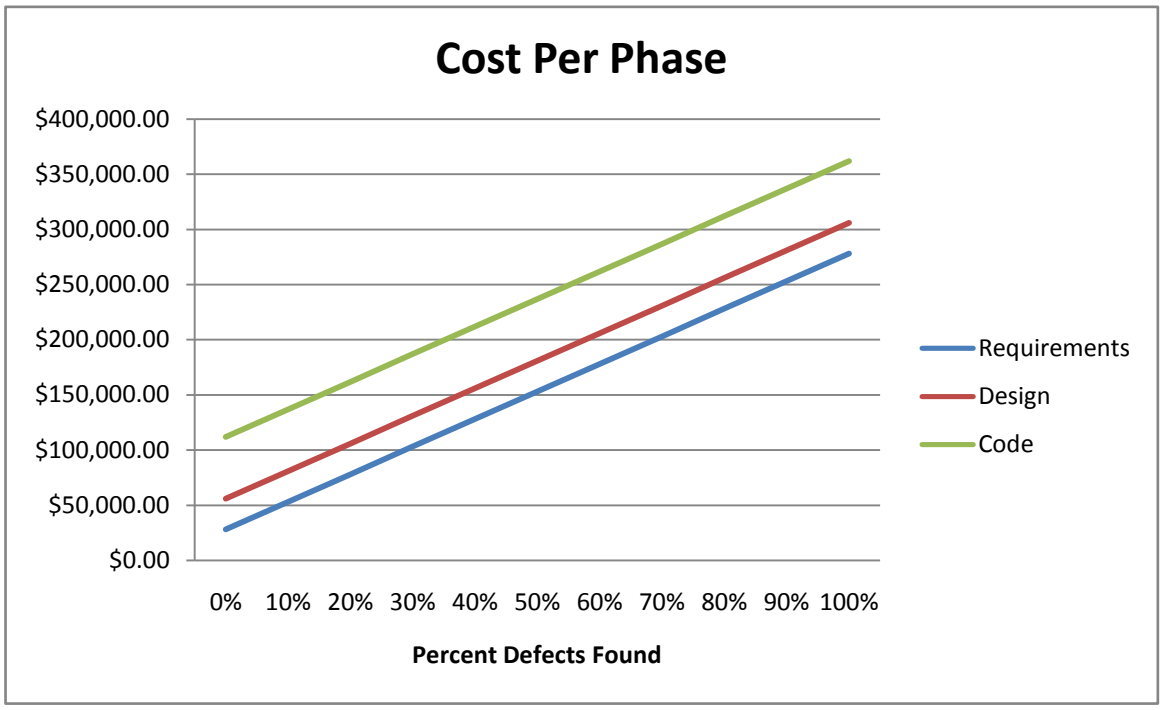

Figure 31 – Cost per phase - $-\frac{1}{2}X + 100$ Detection Rate - 40-80-160 Hour Review

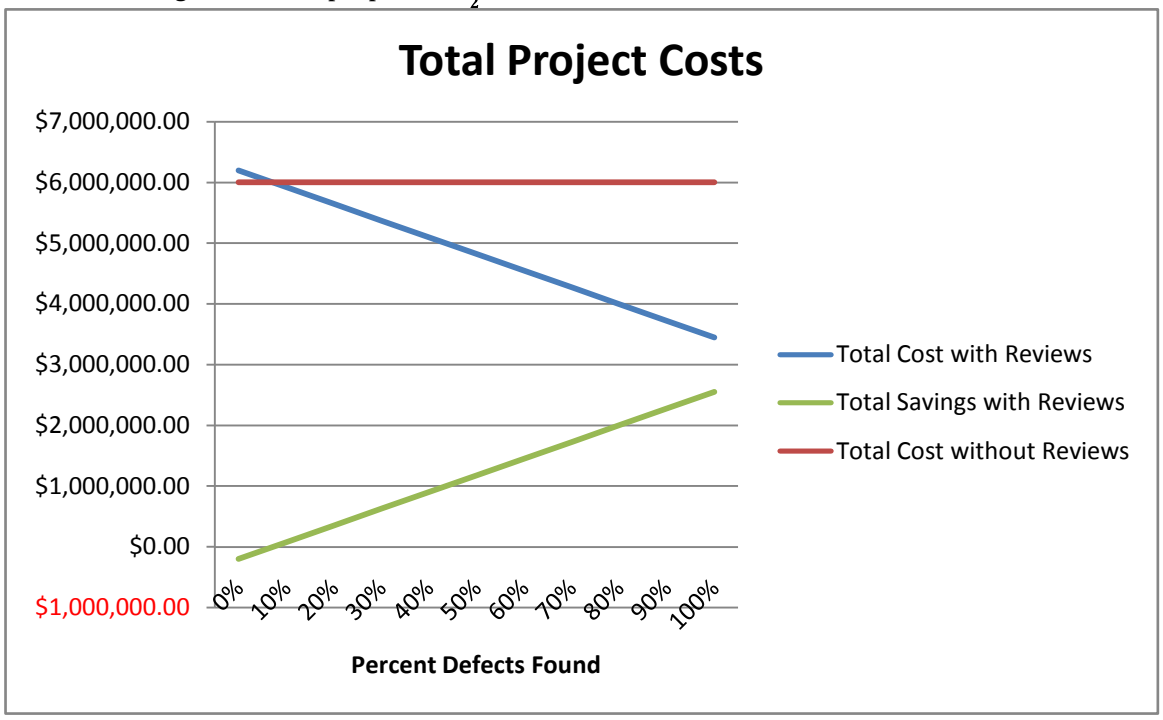

Figure 32 - Total Cost - $-\frac{1}{2}X + 100$ Detection Rate - 40-80-160 Hour Review

The hours per phase rise at the same rate, however as can be seen in Figure 33, they differ significantly from other theoretical analyses. This is because the rate of growth is the same, but since the hours for each review are different the rate is increased as each phase moves forward. The total hours, reflected in Figure 34, shows very little in difference from the other reviews.

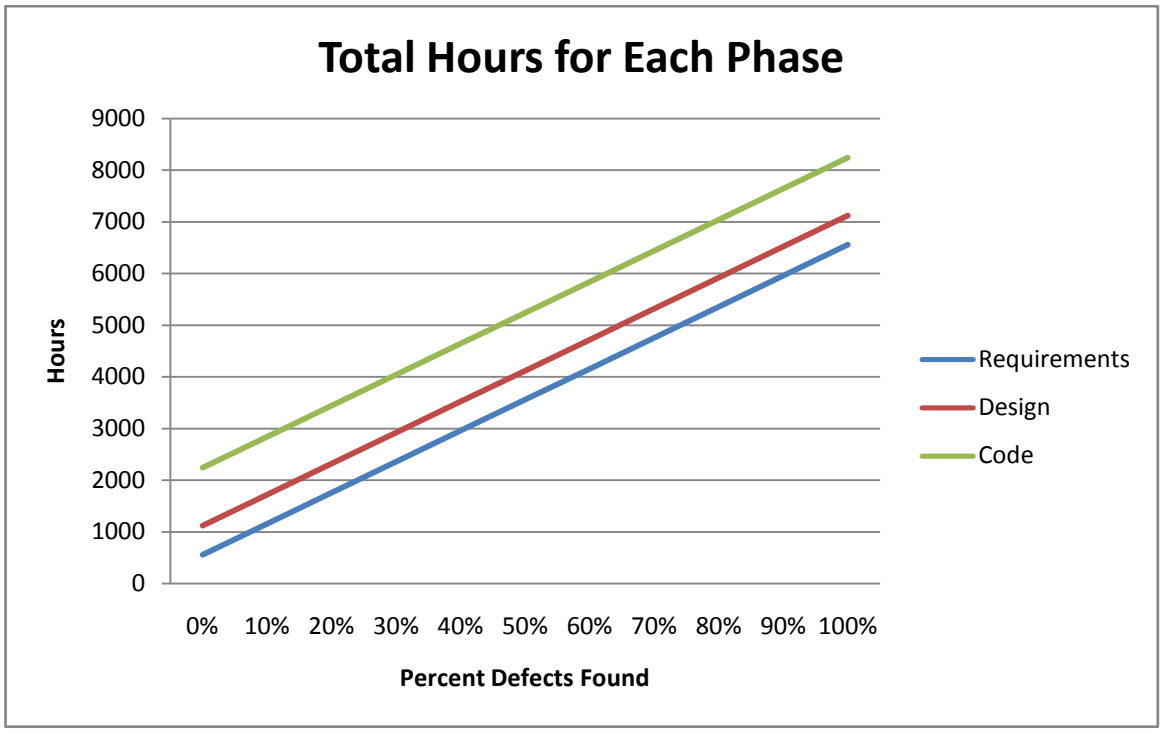

Figure 33 - Hours per Phase - $-\frac{1}{2}X + 100$ Detection Rate - 40-80-160 Hour Review

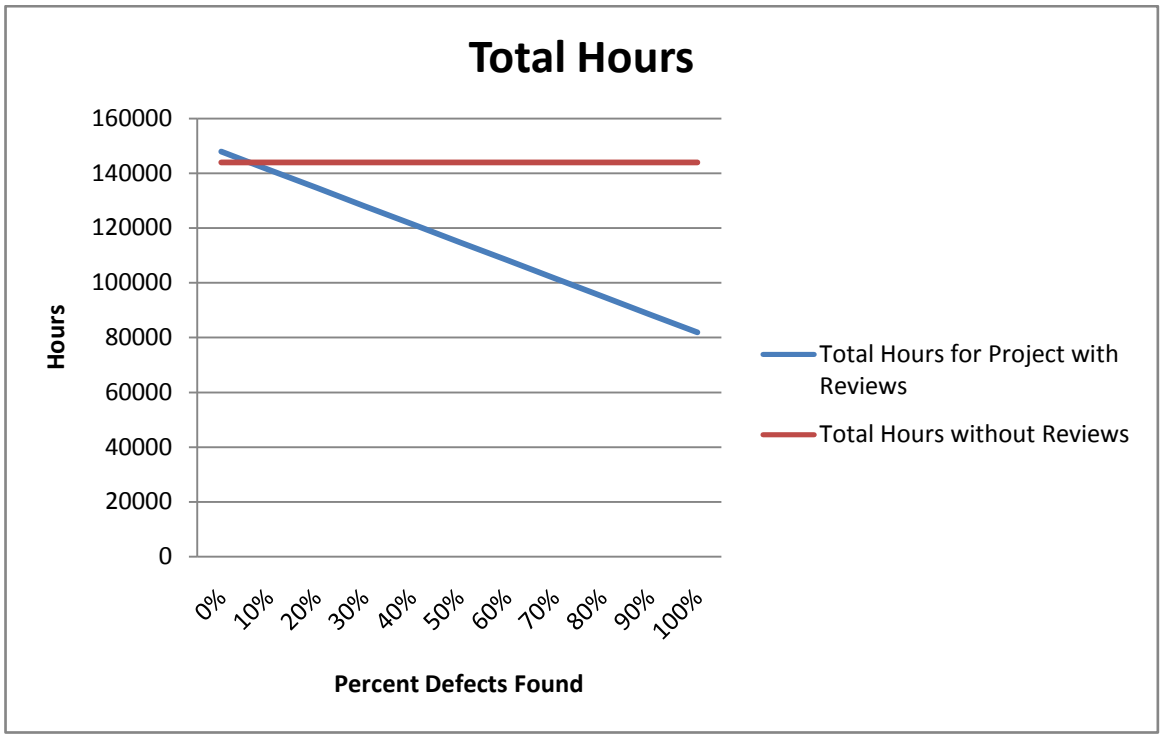

Figure 34 - Total Hours - $-\frac{1}{2}X + 100$ Detection Rate - 40-80-160 Hour Review

6.9–40-80-160 Hr Review Hours - $\frac{1}{2}X$ Detection Rate - 1000 defect

The final theoretical test was conducted using the same 40-80-160 hour review format as the two previous theoretical analyses, however the detection rate was altered to be a $\frac{1}{2}X$ detection rate. This detection rate was 25% – requirements, 50% – design, and 100% – coding. The results are shown below in Figure 35 through Figure 39. The cost per phase was significantly increased, with Figure 35 showing the considerable increase per phase. Figure 36 shows the initial cost was $6,196,000 which again was quickly recouped in the first data point, and the final data point showed a savings of $1,991,500. While this was a little less than some of the other theoretical analyses, it was consistent with the accepted theoretical models. Figure 36 also exhibits the increased difference

between the costs for reviews and total savings with reviews. This increased difference between the phases is the greatest of all the analyses.

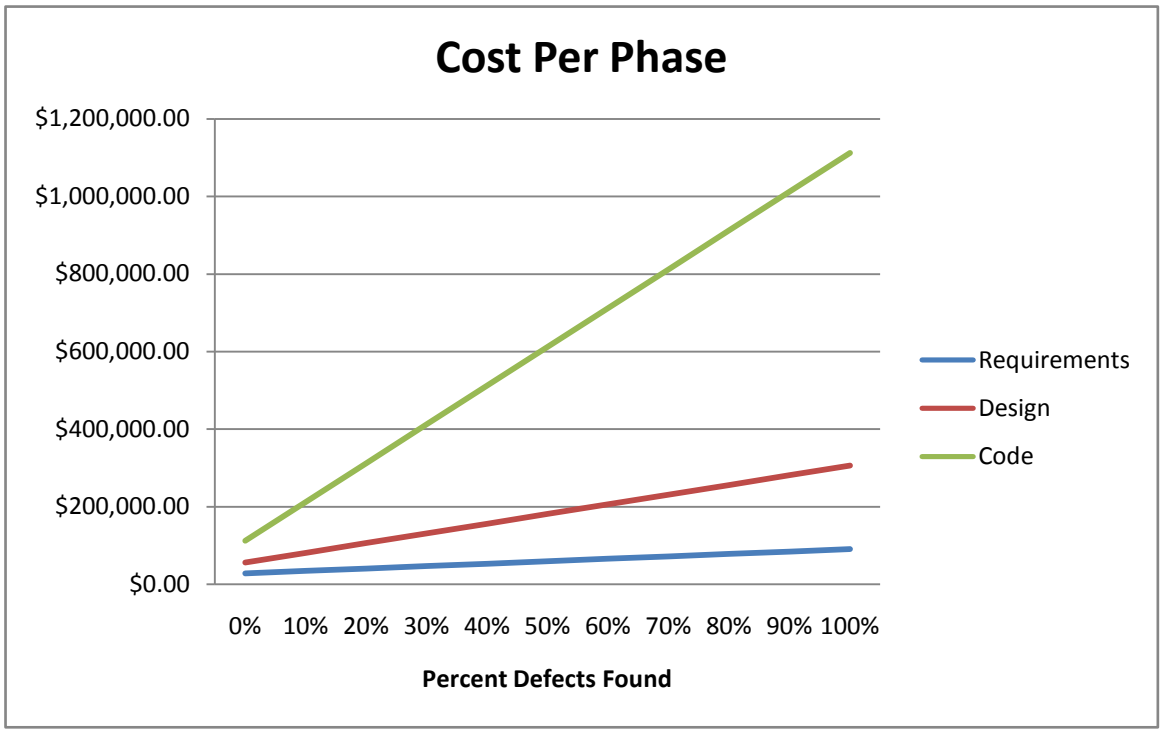

Figure 35 - Cost per Phase -

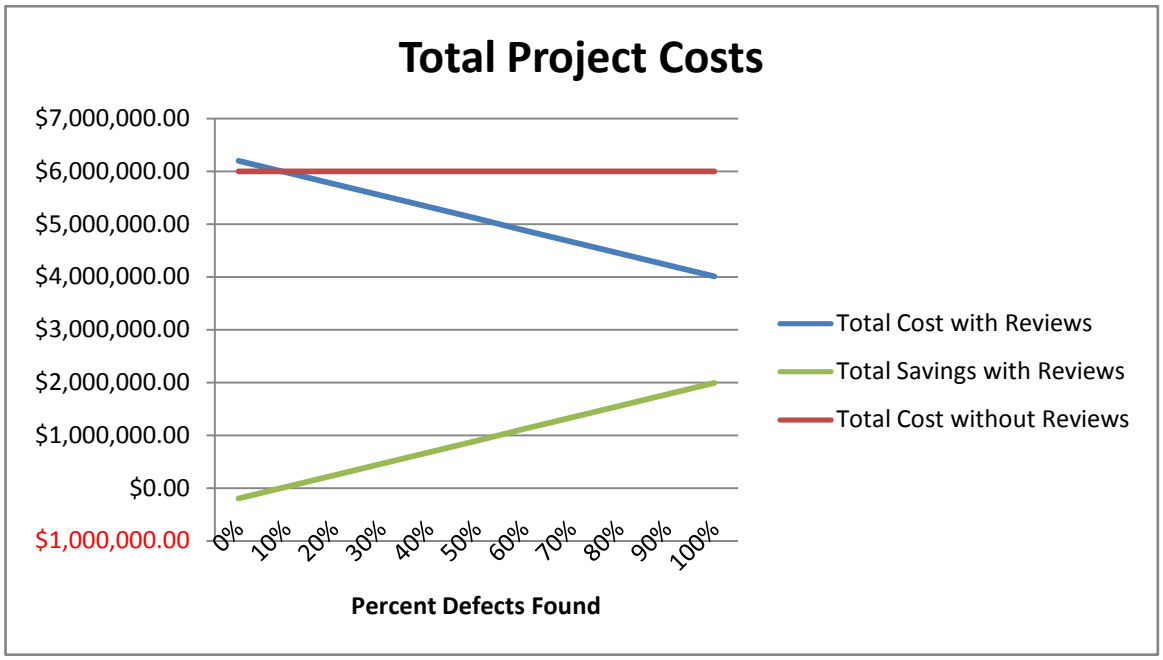

Figure 36 - Total Project Cost

The hours on the project were considerably reduced by using this model, however Figure 37 shows the increase between phases was similar to the cost, and the total hours for the project, as seen in Figure 38, was reduced from the other analyses.

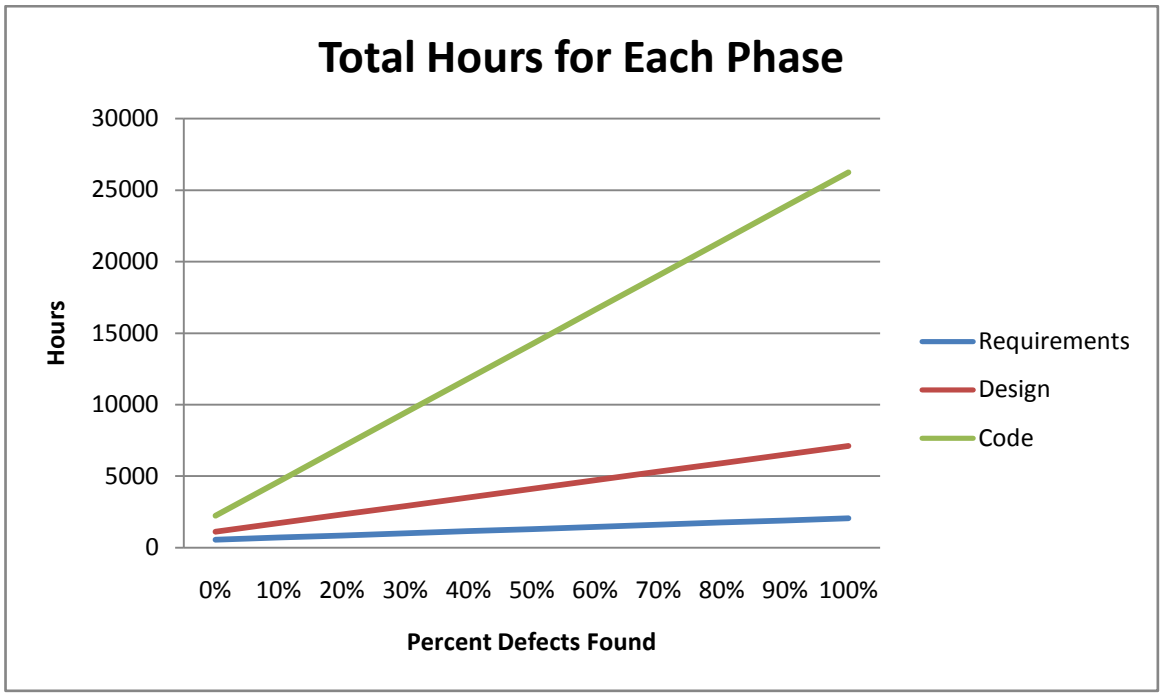

Figure 37 - Total Hours by Phase

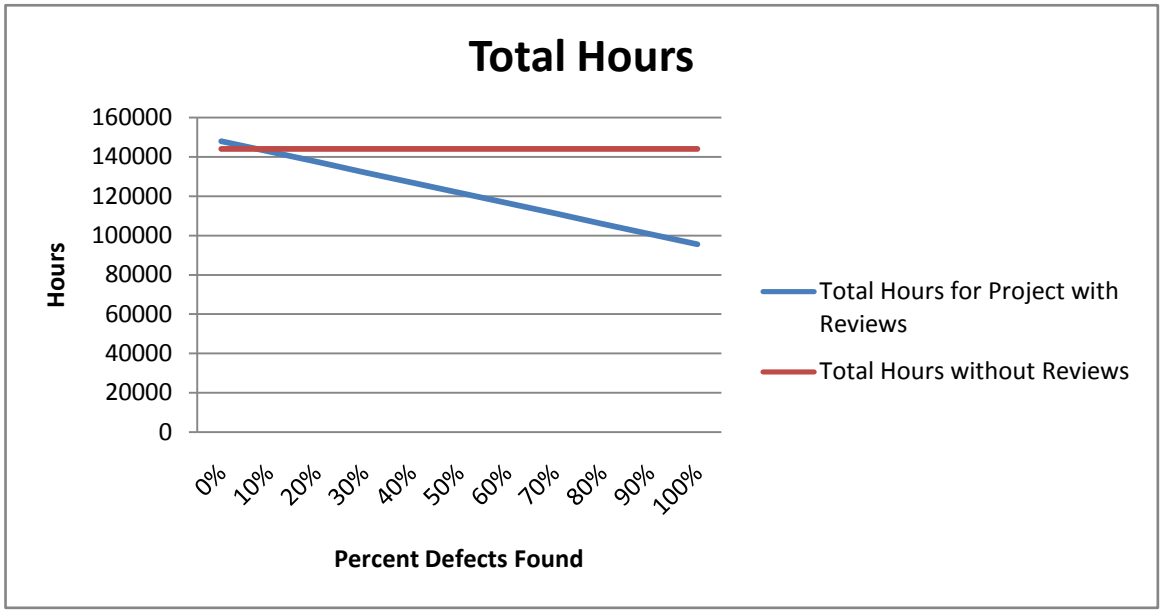

Figure 38 - Total Hours

6.9– Actual Project Analysis

These last graphs are an analysis of actual data from a large project conducted at an aerospace company. The hours for the review are 1/8th of the time required to prepare

for a review, i.e. an 8 hour review requires 64 hours of preparation time, and a 40 hour review requires 320 hours of preparation time for the design phase. The first graph, Figure 39, shows that the number of defects found per hour of review, requirement, design, and code, decreases as the review hours increase. This may be because the easy defects are found quickly and therefore as more time is spent on finding defects they become harder to tease out of the product.

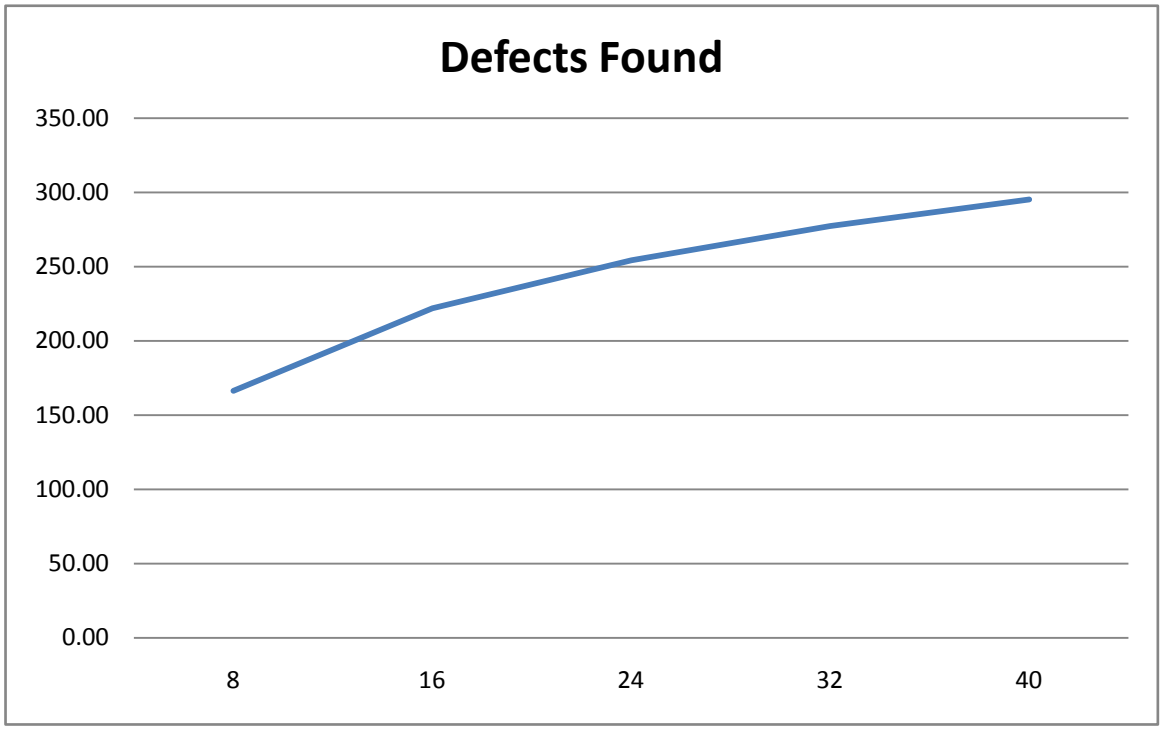

Figure 39 - Defects found

This is consistent with other studies in the information technology field. Jakob Nielsen concluded that "Major usability problems have a higher probability than minor problems of being found in a heuristic evaluation" (1992, p. 380). As shown in Figure 40, he found that as the amount of individuals performed heuristic reviews to determine usability problems, the percentage of defects found degraded and began to have a diminishing return similar to the one shown in Figure 39 (1992, p. 377). Nielson found that

performing a heuristic review with regular usability specialists a team of 3 to 5 was needed to find between 74% and 87% while increasing the number of specialists to 10 only increased the defect detection to about 95% (1992, p. 376). In a theoretical model, this doubling of effort should have doubled the rate of detection, however it doesn't stay the same as illustrated in Figure 39 and through Nielson's work the rate of defect detection diminishes as the amount of effort increases. While Nielson's work does not focus strictly on reviews, it does involve the same challenge that reviews face, such as: differences in personalities, application area experience levels, and project knowledge. A review is ultimately an evaluation of another person's work product and engineering process. There are inherent difficulties in understanding project artifacts created by one or more software engineers, such as technical tradeoffs, experience level with the development paradigm and software engineering tools, and application understanding,. One cannot evaluate a project with the same preciseness that can be applied in civil or electrical engineering as software engineering is a young field yet to achieve scientific underpinnings that support these other engineering disciplines.

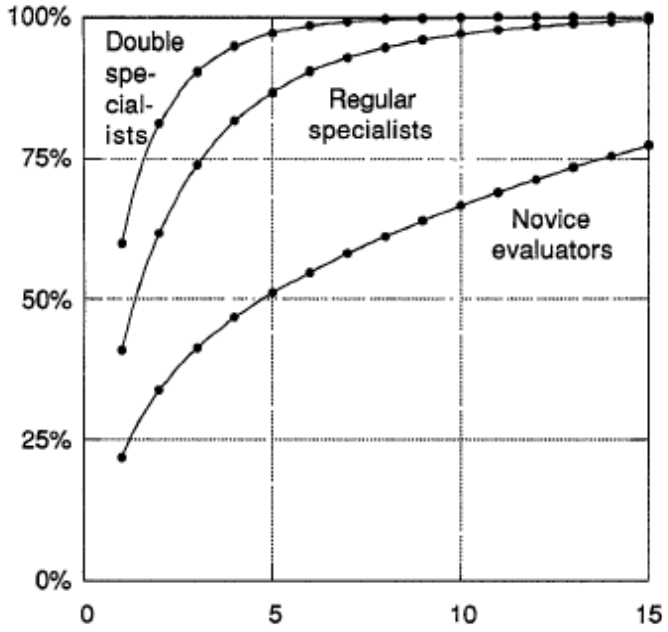

Figure 40 – Heuristic Detection rates

The amount of time spent preparing for the different reviews was significant and increased for the design and coding phase reviews. The design phase 8 hour review took 64 hours to prepare for compared to 32 in the requirements phase, and the review took 128 hours to prepare for during the coding phase. The costs for the 40 hour review were even larger. The 40 hour requirements review took 160 hours and the design review required 320 hours of preparation time. However, the coding review took an astounding 640 hours to prepare for.

Figure 41 - Defect Detection cost per phase

The total costs associated with defect correction showed a diminishing rate of return. The rate of the amount of money associated with the defect correction decreases as the number of hours increases, or as can be seen in Figure 41, the cost rate decreases as the time increases. This is due to the decreasing number of defects found per hour of review.

Figure 42 - Cost per Defect Found

The cost of defect detection, Figure 42, shows a slightly decreasing return on investment. The cost per defect found ranged from $302.97 for an 8 hour review in the requirements phase to an amazing $3,131.03 per defect found in a 40 hour code review. This was an unexpected result that suggests excessive time spent on reviews can be detrimental to the project overall. It is possible to spend too much time on reviews and not get the return expected from the theoretical models. This logically makes sense to anyone who has spent 3-5 days in succession performing code reviews and is consistent with Nielson's results.

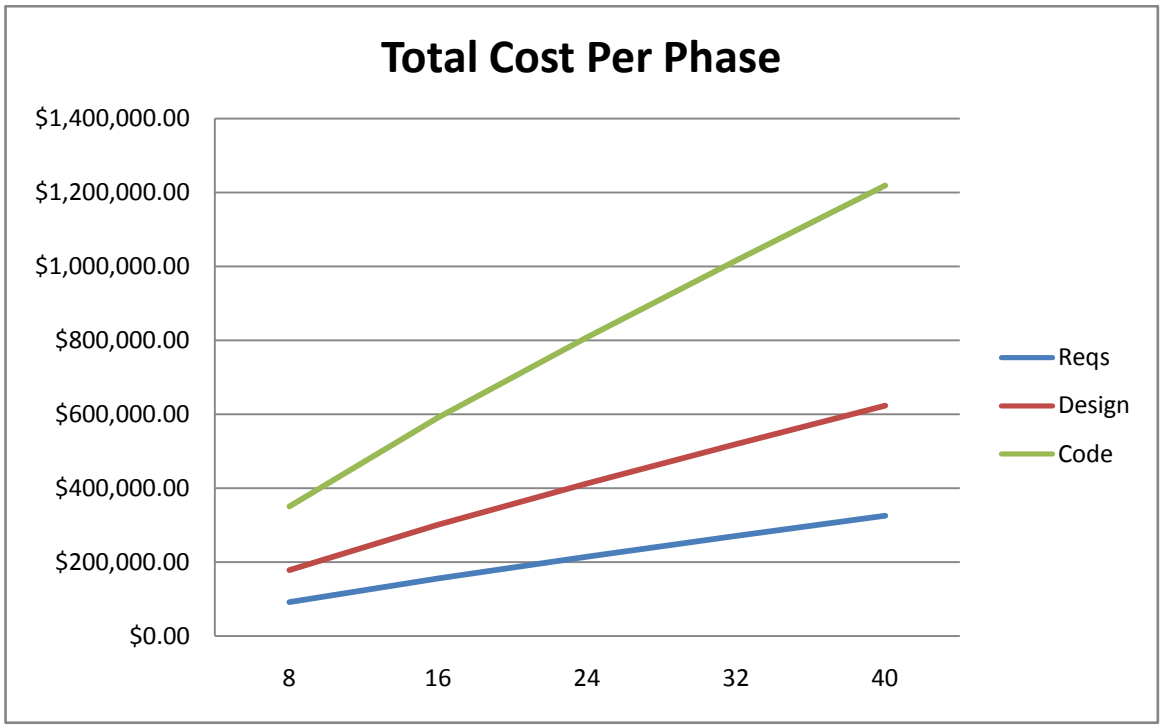

Figure 43 - Cost per Phase

The total cost per phase in a real project is dramatically different from the theoretical model. The total costs associated with the phase are much higher in the real project than in an theoretical project. This cost increase is related to the extra time for preparing for the associated reviews. Anyone who has spent time doing presentations and reviews in a production environment will realize that no one arrives unprepared to this type of meeting. There are numerous hour spent preparing for the review, gathering information, developing the presentation, rehearsing for the presentation, pre inspecting the code to catch obvious errors, and finally ensuring all the data is ready for the presentation. These activities are not limited to code reviews, but also occur in requirement and design reviews. Most people are reluctant to look foolish in front of their peers, so many will go to extraordinary lengths to over prepare for the different

reviews. Reviewers must also prepare answers for other questions. Time and effort is expended developing answers for things such as, product requirements, a brief explanation of the project history and any significant impact it has on the current project culture and design, significant design decisions and what lead to these decisions, and an explanation of the project development strategy. The development of explanations for these types of factors requires significant investment in time and energy.

Chapter 7 – Findings

The theoretical analyses conducted have revealed that the theoretical model fails to show the complete impact of reviews when review time is increased across a project. The theoretical models assume that the rate of error detection and the associated costs will remain the same; however the real world project shows that this is false. As the amount of time associated with each review increases, the models show that the error detection rate decreases. This runs counter to established literature, but makes sense and is supported by Nielson's findings (McConnell, 1993, p. 576). The theoretical models assume that the difficulty in detecting an error is constant, however most developers will agree that this is inconsistent with reality. As a review detects the easily found defects, the number of defects found will decrease because they are harder to detect. Therefore project managers must include this decrease in the frequency of error detection into their calculations when deciding upon the time and effort to assign to a review. As reviewers work to find errors, the rate that they inspect project artifacts (i.e. requirements, design, and code) slows down over time while their ability to detect defects decreases, "the group can overlook errors that would it would otherwise catch" (McConnell, 1993, p. 579).

This is due to the increased difficulty associated with finding additional defect after the initial, obvious, and often simplistic defects are discovered. The other interesting finding is that the amount of time associated with preparing for the review was many times greater than assumed by the author during the initial phases of the theoretical analyses. This increased cost was ignored by the theoretical models which led to an increase in cost savings at a much earlier time in the project.

The average defect detection rate for U.S. software companies is about 85% (Jones, 2008, p. 2). Jones found that on average 77% of requirements defects were removed during a project, and that 85% of defects from the design phase were corrected (2008, p. 2). The defect removal efficiency for the coding phase was 95% and documentation and bad fixes resulted in 80% and 70% respectively (Jones, 2008). According to Jones, "In order to achieve a cumulative defect removal efficiency of 95%, it is necessary to use approximately the following sequence of at least eight defect removal activities:

- Design inspections
- Code inspections
- Unit tests
- New function tests
- Regression tests
- Performance tests
- System tests
- External beta tests

To go above 95 percent, additional removal stages are needed. For example, requirements inspections, test case inspections, and specialized forms of testing, such as human factors testing, add to defect removal efficiency levels" (2008, p. 3). The tests completed above assume that there is 100% defect removal efficiency and this is an unrealistic expectation.

The differences between the theoretical model and the real world project can be related a few specific items. The theoretical model does not take into account the

preparation time associated with the different reviews. The author hypotheses that this increase in preparation time can be attributed to developers wanting to put the best face on their work. The developers will spend considerable time cleaning up the code, ensuring their project is correct. Another factor is that it takes a significant amount of material to conduct a 40 hour review, and this material is typically in a format significantly different than the artifact being reviewed. For example, presentation media might be slides, graphics, and complex mathematical models (i.e. finite state machines, UML diagrams, formulas, and data flow diagrams). This in itself would account for a large portion of the time associated with the review preparation. This review preparation time significantly impacts the real world data but the theoretical model would need to be modified to account for the additional effort associated with the reviews.

Another variable that was not present in the theoretical model was the differences in the error detection rate. The real world project showed a significant decrease in the rate of defect detection per hour, a facto not accounted for in the theoretical models. For the theoretical model to accurately reflect what happens in a real project the error detection rate must be adjusted to account for this finding. The modified theoretical model should also include a way of factoring for the experience of the reviewers. The COCOMO model, originally known as the COCOMO 81, takes developer's experience into account when doing cost estimation, and future iterations of the tool should include these relationships (Sommerville, 2001, p. 524). However, this factor accumulates overall team experience into a single experience factor in the model, a method consistent across the field of Software Engineering at this time. Until more precise methods are

accepted to measure the impact of individual experience on productivity, cumulative methods must suffice.

Chapter 8 – Conclusions

The overall conclusion that can be drawn from this thesis is that while reviews are extremely beneficial, too many reviews requiring too much effort fail to provide improvements in ROI, a key factor in commercial software development. A project manager can conduct reviews, but they must be done judiciously along with other means of error detection to insure a project ships with as few defects as possible. However it is unrealistic to assume that the product will ship with no defects. The consistent rate of return that is projected in theoretical models does not reflect the actual data presented in the real world data considered here. Few project managers would argue that performing reviews were not beneficial, however this study has shown that the amount of time spent on reviews can have a negative impact on the efficiency of those reviews.

From the findings of this study, the optimum time for a review is between 16 and 24 hours. As seen in Figure 39, this time interval allows for the most efficient error detection rate when doing reviews. A project manager who schedules and conducts requirements, design, and code reviews within this window will achieve the best results for that review. If a lesser time window is used, 1-16 hours, then the reviewers will leave an unacceptable amount of defects in the product, but if a larger time frame is implemented, 24-40 hours then the project will see diminishing returns on the investment of additional time. This diminished return may be acceptable in safety critical systems but may become fiscally wasteful in COTS or custom fixed price software projects.

Future extensions of this work should include a modification of the data detection tool to account for the experience of the developers, a weighted scale for the preparation

time that increases as the preparation time increase and decreases as the preparation time decreases, a more detailed error detection rate that accounts for the decreasing detection rate found above, and a variable that takes into account the change in frequency of defects found as the reviewers spend more time in the review. Another extension of this thesis would be to gather and analyze additional data from other real world projects then compare them to this experiment, with an emphasis on determining if the type of project affected the outcome. A researcher could then develop ratios to account for the missing variables in this study. The development of an accurate tool to estimate the real cost of reviews would then be possible, thereby allowing management to accurately estimate the cost of requirements, design, and coding reviews.

Another field of study that would enhance this work, is to research the manufacturing engineering disciplines to determine of any of the theories from zero defects in manufacturing could be applied to these models. The application of these accepted design practices from other engineering disciplines may have merit and be usable for this study. However, many engineering fields focus quality efforts on reproduction of the design rather design production measurements.

Bibliography

Ahern, D. M., Clouse, A., Turner, R., (2008) *CMMI Distilled: A Practical Introduction to Integrated Process Improvement*, Addison-Wesley Professional.

Bureau of Labor Statisics, U.S. Department of Labor. (2007, December 18). *Computer Software Engineers.* Retrieved December 22, 2007, from http://www.bls.gov/oco/ocos267.htm

Fagan, M. E. (1976). Design and code inspections to reduce errors in program development. *IBM Systems Journal , 15* (3), 258-287.

Henry, J. (2003). Software Project Management - A Real-World Guide to Success. Boston: Pearson Addison Wesley.

Jones, C. (2008). Measuring Defect Potentials and Defect Removal Efficiency. *Crosstalk - The Journal of Defense Software Engineering* , 1-5.

McConnell, S. (1993). Code Complete. Redmond, WA, USA: Microsoft Press.

Nielson, J. (1992). Finding Usability Problems Through Heuristic Evaluations. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (pp. 373-380). Monterey, California, United States: ACM.

Porter, A., Siy, H., Toman, C. A., & Vota, L. G. (1995). An Experiment to assess the cost-benefits of code inspections in large scale software development. *SISGOFT* (p. 92.103). Washington: ACM.

Pressman, R. (2001), Software Engineering: A Practitioner's Approach, New York, McGraw Hill.

Sommerville, I. (2001). Software Engineering (6 ed.). Essex, England: Pearson Education Limited.