Spring 2018

Choosing the Best Software for Spec Review: An Analytic Hierarchy Process Approach

Bob Greenlee
Portland State University

Portland State University

Maseeh College of Engineering and Computer Science

Department of Engineering and Technology Management

Choosing the Best Software for Spec Review

An Analytic Hierarchy Process Approach

Bob Greenlee

ETM 530 – Decision Making (Spring 2018)

Dr. Ramin Neshati

June 6, 2018

Abstract

Every manufacturing company is faced with the problem of accurately converting customer requirements into products and deliverables, including a thorough review of customer specifications. Software can help facilitate this tedious and time consuming process, but choosing or designing the best solution is difficult due to conflicting and poorly prioritized criteria.

This paper illustrates the use of an analytic hierarchy process, specifically the hierarchical decision model (HDM) tool developed at Portland State University, to select the best software solution for facilitating customer spec review. In addition to resolving the primary question of which software to use, it shows how the HDM tool can be used to rank the selection criteria in a way that is useful for communicating requirements to the software developers.

This study suggests that an analytic hierarchy process may be useful for prioritizing criteria for other design problems, and suggests improvements for the HDM tool.

Contents

Do I contradict myself?

Very well then I contradict myself,

(I am large, I contain multitudes.)

-Walt Whitman, *Song of Myself*

BACKGROUND & INTRODUCTION TO THE DECISION PROBLEM

Conversion of customer requirements

Every company has multiple processes that involve the flow of information to and from customers. Particularly important are those processes that convert a customer's stated requirements into information used to produce and deliver what they expect. In a manufacturing company, these include processes for:

- Providing a quote to a customer
- Reviewing and booking a purchase order
- Reviewing and clarifying customer data with regard to manufacturability
- Reviewing customer specifications when they are different from or additions to industry standards
- Flow down of customer requirements to the various departments and subcontract suppliers that need them
- Requesting deviations from customer requirements

Clear and accurate information flow is vital to the success of a company that seeks to put the needs of its customer first. As important as these processes are, they can be tedious, time-consuming, and boring for even the most detail-oriented person, and many require a deep understanding of specifications, manufacturing capability, and customer perspective in order to accurately translate and ultimately deliver what the customer has ordered.

When the flow of information crosses between functional groups within a company, the translation problem can become a game of telephone, where one group (e.g., Customer Service) says one thing, while another group (e.g., Front End Engineering) hears something entirely different. These miscommunications can be extremely costly when the customer does not get what they wanted, or more is delivered than was needed.

Even more challenging is the situation where cross-functional groups within a company must work together to review a customer's requirements and then provide coherent feedback to a customer. Particularly when the designated members of these different groups have competing demands on their time, carefully reading a legal document and writing out a list of exceptions or questions can easily become the last item on a long list of priorities.

The focus of this paper is on the best way to facilitate a cross-functional review of customer specifications within a manufacturing organization. Because these specifications can become part of a binding contract, it is important that they be thoroughly evaluated to determine whether the organization can meet them, and to ensure that they are properly quoted.

Customer spec review

The context in which this study was conducted is a medium size (~500 employee) high-end printed circuit board company that relies on customer data and specifications for manufacturing (known as a "build to print" operation). In this company, the specification review process was once conducted in meetings with an attendee from each functional group reading the spec line-by-line, raising issues, questions, or concerns which were then captured in a document and sent to the customer for feedback. These meetings were not well attended, and eventually the onus of the spec review fell on a Front End Engineering representative, who was well-versed in the specs and who could request clarification via emails from other functional groups if they had questions. This worked well for a time, but without additional input from other areas, such as Process Engineering, Quality, Shipping, or the metallurgical lab, there was a risk that some items would be missed.

Better living through software?

Perhaps just as importantly, it was felt to be unfair to put the burden of spec review on one individual, so this time-intensive process was replaced with a streamlined process that involved sending the specification via email to a “Spec Review” email distribution list. Representatives from each area were expected to send their feedback on the customer spec to the group, either stating that there were no issues, or writing any issues or concerns on a form that would then be forward by Customer Service back to the customer.

Needless to say, when people are busy, responding to an email request to review a spec may not be everyone’s highest priority. Without some way for management or peers to see who had and who had not responded to the spec, it was easy for a reviewer to ignore the email or simply respond with “No issues” after a cursory reading or it. Because in some cases booking a purchase order (PO) depended on a timely review and approval of the specification, specs were sometimes accepted after only one or two functional group approvers had chimed in.

I approached our IT group with this problem, and suggested that we might use an issue-tracking software program called Jira to track specifications as they went through the review and approval process. Jira allowed specifications and other documents to be attached to the request, and simple workflows allowed the spec to go from one approval group to the next, or to be sent to multiple approval groups simultaneously. The initial draft of the workflow tended to bombard the person submitting the spec (generally the customer service representative) with emails alerting them to the fact that someone had approved, or someone had submitted an exception request. It also bombarded the reviewer who had not yet gotten around to reviewing the spec with reminder emails. Unfortunately, this did not go over well with the Customer Service Manager, and the spec review returned to the less-than-adequate email approach.

Tracking each functional group's approvals became a critical issue after a recertification audit, because it was difficult to show verifiable, objective evidence that a specification had been reviewed and approved. An Excel spreadsheet was used to keep track of which spec had been submitted and the date that it was "put online in the spec hub" (i.e., stored on a network location), but there was no way to see who had reviewed it or when.

Even worse, customers often provided specifications that were conditional, and only applied if they were called out on a PO. These so-called quality "Q-codes" received only a cursory review, since it was generally understood that they would get a more thorough review if and when they were specified on the purchase order. Other documents, like terms and condition (T's and C's), environmental requirements, and even email communications from the customer gradually made their way into the customer spec hub, making it very difficult to tell which specs were related to manufacturing, which were related to packaging and shipping requirements, and which specified the paperwork, test coupons, and other deliverables the customer wanted with their printed circuit boards. Specifications were grouped by customer name, even though different divisions could have different specifications. In a word, the spec hub was quickly becoming disorganized and hard to use.

Specifying Spec Review Software Requirements

When one asks a software designer, “What do you think it is possible for this software to do?” a common response is, “We can make it do anything you want it to.” The problem, of course, is in defining exactly what you want the software to do. I approached the IT department again with the problem, only this time did not presuppose a solution, but instead proposed an analytic approach to the question of what software would be the best choice for the spec review process. The strength of the HDM approach is that it could provide a clear ranking of criteria which could then be used to prioritize the design of the software.

This project was a good example of “thinking fast and slow” (Kahneman, 2011), though sometimes it felt like a race between management, who wanted to decide on a solution as soon as possible based on intuition and gut feel, and engineering, who wanted to ensure that the solution was as robust and useful as possible. In this case both runners won, since the solution that management was leaning toward was confirmed and further refined by the hierarchical decision model.

METHODOLOGY OVERVIEW

Decision Making Heuristics are Fast but Lack Discernment

Because there was some urgency in creating a robust solution to the spec review problem, quality management was inclined toward a quick evaluation of existing options. These included two off-the-shelf solutions: Atlassian's Jira bug tracking software, Microsoft's SharePoint software, and a hybrid of existing custom software developed in-house for managing customer data reviews and buildability reviews. The incumbent solution, using Microsoft Outlook to email attachments coupled with a Microsoft Excel spreadsheet to show the approval status, was considered inadequate for tracking approval data required by system auditors. The use of voting Outlook voting buttons to track approvals was briefly considered as a possible short term solution, but this was not seen as very user friendly.

After a quick test of the Jira solution, which resulted in a large number of unwanted "reminder" emails, Management turned toward existing custom solutions that could possibly be adapted for the spec review process. A quick demonstration of these approaches revealed several desirable features including a dashboard for tracking progress, pre-populated drop-down choices for providing feedback to customers, and a clean, simple user interface. Based primarily on these quick reviews, management sketched out a short list of criteria for the new software and emailed it to the IT department, the assumed experts in making good software.

While an intuitive "take-the- best" heuristic (Gigerenzer & Goldstein, 1996, Gladwell, 2005) such as this may make some sense when a quick decision is required, it is not particularly helpful when it comes to selecting and prioritizing specific criteria for the solution. A great deal of decision making literature (e.g., Kahneman,2011, Moxley, Ericsson, Charness, & Krampe, 2012, Milkman, Chugh, & Bazerman, 2006) supports the value of intuition, but also insists that intuition can be improved by deliberative decision making processes. Likewise, polling one or two quasi-experts tends to create a bias in favor of particular software features without a clear understanding of what the users of that software might deem most important. With this in mind, I decided to take another approach and attempted to apply the hierarchical decision model to the problem of selecting the best spec review software.

Hierarchical Decision Modeling

The hierarchical decision model (HDM) was developed by Portland State University's Dundar Kocaoglu in 1976. It is a web-based tool that follows the analytic hierarchy process (AHP) (Saaty, 2008), and involves three essential steps (Salgado, E., Salomon, V., & Mello, C., 2012):

1. Identification of decision criteria and alternatives.
2. Designation of weights for the criteria and priorities for the alternatives.
3. Synthesis of the results.

Experts are asked to make pairwise comparisons of sometimes intangible criteria using a 1 – 99 point scale. This apportionment of points allows for a ranking of the criteria, and the tool produces aggregate statistics for each expert that can then be analyzed and synthesized to help make complex decisions manageable and defensible.

AHP methods have recently been applied to new product development (Salgado, Salomon, & Mello 2012; Yuen, 2014; Tu, Zhang, He, Zhang, & Li, 2011), with mixed results largely due to the complexity and time commitment required from experts inherent in this approach. Still, the fact that it can successfully prioritize customer requirements suggests that it is a good fit for the often complex and conflicting criteria presented to IT departments when asked to develop new software.

In order to develop the HDM model for selection of spec review software, we first had to determine the possible software options and then select criteria based on different perspectives.

SELECTING AND VALIDATING POSSIBLE SOFTWARE OPTIONS

The initial set of software solutions included the current solution (email/spreadsheet), three off-the-shelf solutions (an Oracle GSM module, a Sharepoint solution, and a Jira solution), and a custom solution based on existing in-house software solutions for other processes. After discussions with the software manager in charge of the Oracle database, I realized that the resources for supporting this new module would not be available, and so I removed it from consideration. Likewise, the existing solution was considered by users and management alike to not be meeting the needs of the spec review process. Ultimately, the remaining three solutions (Jira, Sharepoint, and custom) were chosen as potential options for the HDM model.

SELECTING & VALIDATING HDM PERSPECTIVES AND CRITERIA

Initially, only the perspectives of management and a representative from each of the functional groups who would use the software were considered. These were gathered by informal interviews with representatives from each group, and compiled (see Table 1).

Perspective	**Criteria**
Customer Service	Quicker response, simpler issue resolution with customer, improved spec organization, quick identification of showstoppers
Tooling	Accuracy, completeness, summary of key issues, more thorough review, include review of Q-codes
Quality	Better approval traceability, captures special

	quality system, first article, and C of C requirements
Shipping	Captures deliverables, eliminates need for "cheat sheets", ties specs to specific order
Engineering	Each area only reviews specs that pertain to them, specs have a clear tie to manufacturability guidelines, specs have a clear tie to in-process test specs
Management	clear responsibilities, easy to use, low cost, readily available

Table 1. Initial list of perspectives and criteria.

While viewing the problem from the perspective of each approval group helped with brainstorming criteria for the model, it neglected one important group entirely, our Information Technology (IT) group who would be responsible for developing the software. It also neglected the fact that there was substantial overlap in the criteria that users in each functional group shared. I added criteria relevant to the IT department to the list, and created an initial HDM model (see Figure 1).

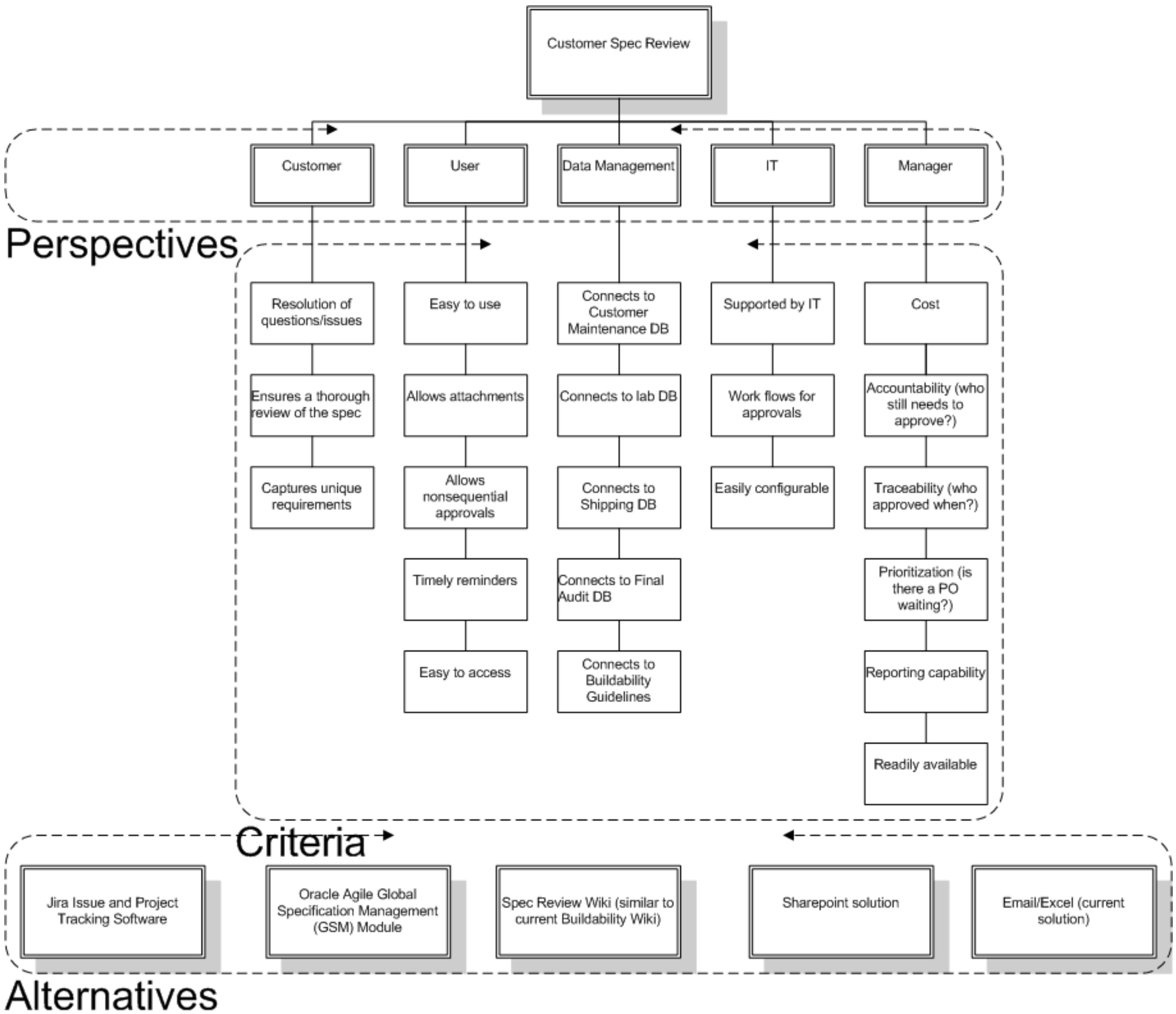

Figure 1. Initial HDM model for the Customer Spec Review process.

A simple calculation revealed that two hundred seventy-one pairwise comparisons would be required by this model, and that it would take about forty-five minutes to complete if the expert took ten seconds to consider each choice.

In a second approach to the model, I grouped the criteria by the desired high-level features of the software: functionality, traceability, availability, ease of use, and control. This helped bring out additional criteria such as revision control, security, spec organization, and access to prior revisions of the specs. However, this second draft promised to be even more challenging for experts to complete, with a total of two hundred ninety-nine pairwise comparisons, or roughly fifty minutes worth of comparisons. After a test drive of the model, it was clear that a simpler model had to be developed.

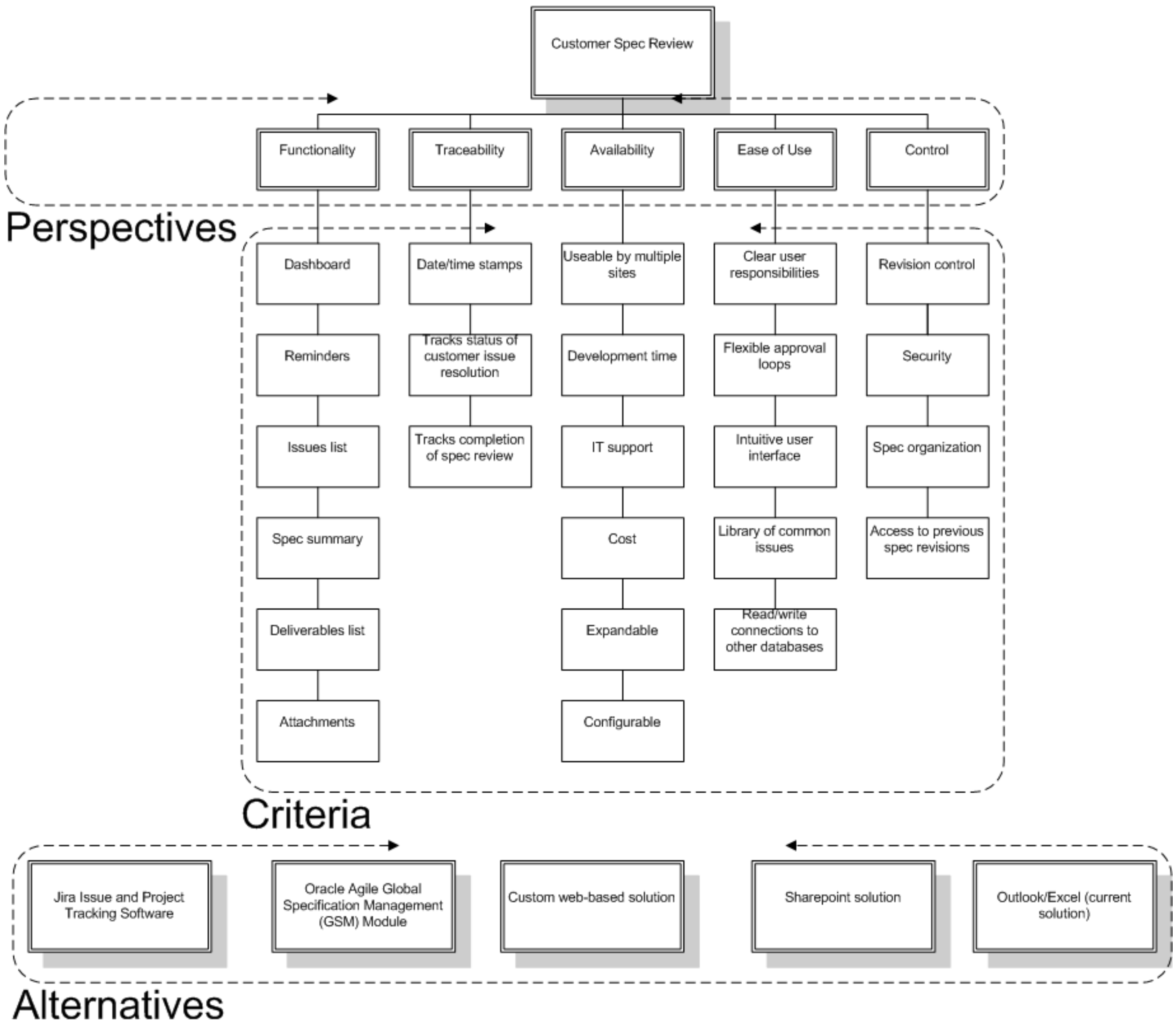

Figure 2. Second draft of HDM model for the Customer Spec Review process.

By grouping criteria under three major perspectives (Management, IT, Users), eliminating the two least likely alternatives (the unsupported Oracle module and the unacceptable current Outlook/Excel solution), and removing what appeared to be confusing criteria (e.g., “expandable” and “configurable”) or criteria that were standard across all solutions (e.g., security and revision control), I was able to trim the model down to a manageable eighty-seven pairwise comparisons, which would take approximately fifteen minutes to complete (see Figure 3).

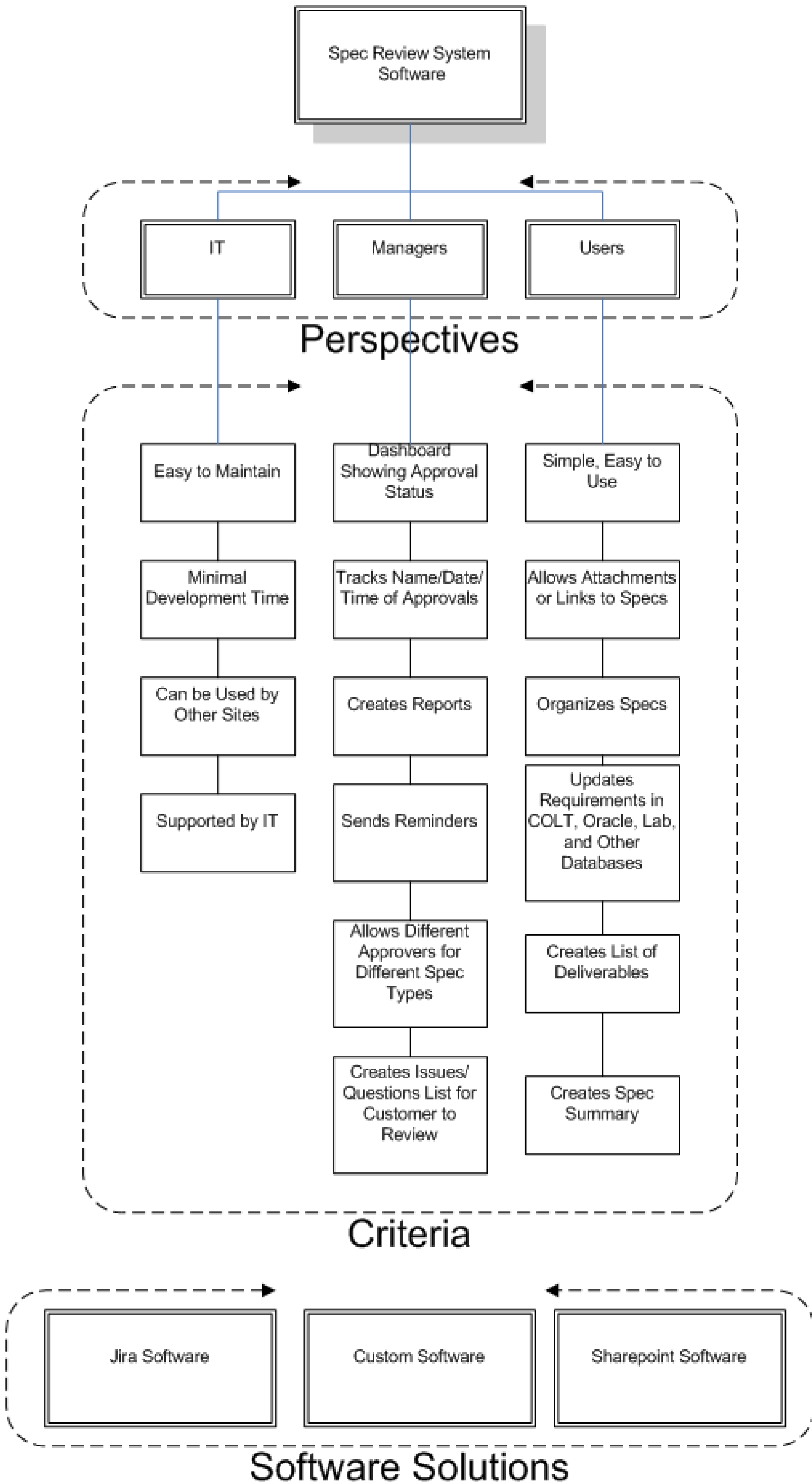

Figure 3. Final simplified HDM model for the Customer Spec Review process.

This turned out to be a wise decision, since every one of my experts was busy and in high demand. Given that it was already difficult for these individuals to find time to review customer specifications in detail (this was part of the problem), convincing them to take forty-five or fifty minutes to complete a survey comparing vague software criteria bordered on nearly impossible. Indeed, even fifteen or twenty minutes of their time was seen as a significant intrusion. Recent behavioral economic literature (e.g., Leigh, 2015) encouraged me to employ additional strategies to help improve participation, including proper framing ("This will help us develop a better spec review software solution"), bribery ("If you act now, you could win a new car!"), guilt ("Since I just finished your customer pre-audit survey, do you think you could start on mine?"), and one-on-one cajoling, with the promise of treats. As will be seen, these were for the most part effective.

DATA SOURCES FOR MODEL QUANTIFICATION

I identified individuals from each perspective (manager, IT, and user), and reached out to them with an email describing the project and goals (see Table 2).

Perspective	Expert
IT	K.B., Corporate IT Director (responsible for custom in-house software solutions) **M.P**., Corporate Sharepoint Developer **A.K.**, Corporate IT Manager (responsible for Oracle and Jira software solutions) **K.S**., Software Developer **R.R.**, Document Control (Sharepoint superuser)
Management	**A.P.**, Quality Manager (owner of spec review process) **E.W.**, Front End Engineering Manager S.D., Process Engineering Manager
User	**S.B.**, Front End Engineering Technician (responsible for managing current spec review system) **E.N.**, Application Engineer **V.H.**, Customer Service Representative

	T.R., Customer Service Representative (results later removed due to inconsistency) S.B., Lab Supervisor **B.G.**, Quality Engineer

Table 2. HDM experts (**bold** indicates completion of survey).

Of these fourteen experts, eleven (including myself) completed the HDM survey, with good representation from each of the three perspectives.

Quantification Process

It should be clear from this list that there are significant differences in the levels of expertise among the responders to the survey. IT professionals understand what the capabilities and limitations of each software option, but are not necessarily as well-versed in the needs of management for data, or the frustrations of users with a cumbersome software program. And not all users are familiar with the possibilities custom software option could provide, so when asked I provided some informal assistance in the form of high-level descriptions of each software approach.

Overall Results

The overall results (see Figure 4) were not surprising: custom software had already been developed for reviewing customer data and automatically creating a list of technical queries, and there already existed other custom software which had the capability of facilitating engineering's review of customer designs. The pairwise comparisons of the criteria confirmed the intuition that a hybrid of existing custom software would provide the best solution to the spec review problem.

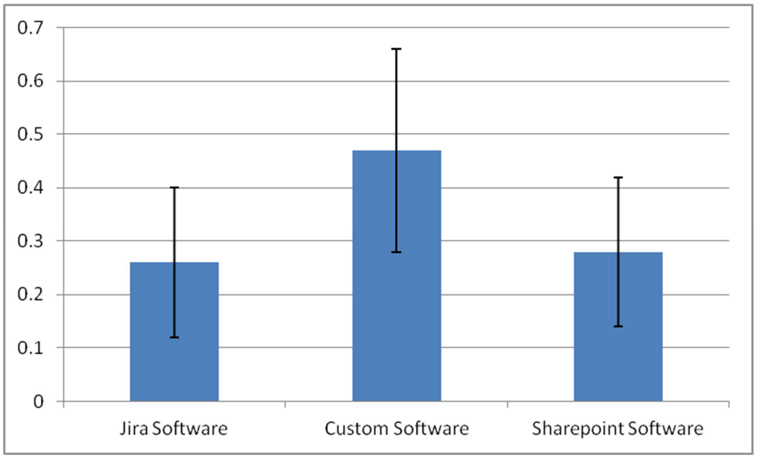

Figure 4. Results of HDM software selection survey (mean ±1 σ)

What was somewhat surprising was that the F-test value was below the critical F-value at the p = .05 level (but just barely; see Figure 5 below), implying that the confidence level of this result was just barely below 95%.

Source of Variation	Sum of Square	Deg. of freedom	Mean Square	F-test value
Between Subjects:	0.27	2	.133	3.48
Between Conditions:	0.00	10	0.000	
Residual:	0.76	20	0.038	
Total:	**1.03**	**32**		
Critical F-value with degrees of freedom 2 & 20 at 0.01 level:				5.85
Critical F-value with degrees of freedom 2 & 20 at 0.025 level:				4.46
Critical F-value with degrees of freedom 2 & 20 at 0.05 level:				3.49
Critical F-value with degrees of freedom 2 & 20 at 0.1 level:				2.59

Figure 5. Results of the statistical F-test for evaluating the null hypothesis

Inconsistency & Disagreement

One did not have far to go to see the source of this barely failing F-test value, as the HDM tool reported that the major source of the variation was between subjects, and the tool also reported inconsistency by expert (see Figure 6). All but one expert had inconsistency scores below 0.04, with the one exception exhibiting a relatively high 0.13 level of inconsistency. Not only did this expert rank the Jira software option much higher than the other two options on every paired comparison, they also inconsistently ranked these options on the majority of the paired comparisons, resulting in an inconsistency score of 0.26 for the level 2 criteria, and an overall inconsistency of 0.13. Rather than request this person repeat the survey (they had initially been reluctant to complete it due to time constraints), I chose to remove their results from the model.

Spec Review System Software	Jira Software	Custom Software	Sharepoint Software	Inconsistency
A	0.25	0.46	0.28	0.03
A	0.17	0.6	0.23	0.01
B	0.3	0.4	0.3	0.01
E	0.33	0.33	0.33	0.01
E	0.14	0.67	0.19	0.01
K	0.07	0.86	0.07	0
M	0.32	0.37	0.32	0.01
R	0.17	0.23	0.59	0.04
S	0.2	0.53	0.27	0.02
T	0.61	0.29	0.11	0.13
V	0.29	0.31	0.4	0.01
Mean	**0.26**	**0.46**	**0.28**	
Minimum	**0.07**	**0.23**	**0.07**	
Maximum	**0.61**	**0.86**	**0.59**	
Std. Deviation	**0.14**	**0.18**	**0.13**	
Disagreement				**0.125**

Figure 6. Individual and aggregate results of the HDM model.

Removal of this expert's results did not change the overall outcome, as seen in Figure 7, but did dramatically improve the F statistic (see Figure 8), suggesting that these results were quite significant. Note, however, it did *not* dramatically reduce the disagreement statistic, which had been 0.125 and was now 0.115.

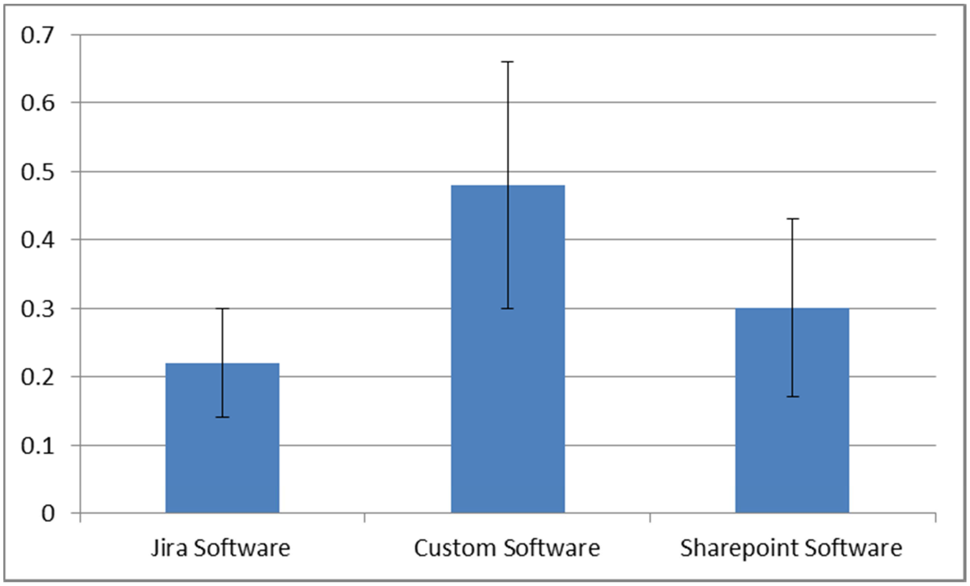

Figure 7. Results of HDM software selection survey (mean ±1 σ) after removing T. R. data.

Source of Variation	Sum of Square	Deg. of freedom	Mean Square	F-test value
Between Subjects:	0.34	2	.168	5.36
Between Conditions:	0.00	9	0.000	
Residual:	0.56	18	0.031	
Total:	**0.90**	**29**		
Critical F-value with degrees of freedom 2 & 18 at 0.01 level:				6.01
Critical F-value with degrees of freedom 2 & 18 at 0.025 level:				4.56
Critical F-value with degrees of freedom 2 & 18 at 0.05 level:				3.55
Critical F-value with degrees of freedom 2 & 18 at 0.1 level:				2.62

Figure 8. Results of the statistical F-test for evaluating the null hypothesis after removing T.R. data.

Even though the experts were, for the most part, internally consistent, there was still significant disagreement on the ranking of the criteria. This made some sense, especially since the software experts recognized that each option is capable of customization, and since each expert had different levels of experience with each option. Rather than attempt to reconcile the disagreement, which at 0.115 was just above the generally accepted level of 0.10, I chose to take it as indicative of the variety within my selection of experts who represented different perspectives (IT , Management, and Users), and so could be expected to weight the criteria differently.

Spec Review System Software	Jira Software	Custom Software	Sharepoint Software	Inconsistency
A	0.25	0.46	0.28	0.03
A	0.17	0.6	0.23	0.01
B	0.3	0.4	0.3	0.01
E	0.33	0.33	0.33	0.01
E	0.14	0.67	0.19	0.01
K	0.07	0.86	0.07	0
M	0.32	0.37	0.32	0.01
R	0.17	0.23	0.59	0.04
S	0.2	0.53	0.27	0.02
V	0.29	0.31	0.4	0.01
Mean	**0.22**	**0.48**	**0.3**	
Minimum	**0.07**	**0.23**	**0.07**	
Maximum	**0.33**	**0.86**	**0.59**	
Std. Deviation	**0.08**	**0.18**	**0.13**	
Disagreement				**0.115**

Figure 9. Individual and aggregate results of the HDM model after removing T. R. data.

Perspectives & Decision Criteria Weights

The average weights for each perspective and decision criteria are shown in Figure 10. The user's perspective was generally seen as the most important, and of the criteria grouped under the user perspective, "Creates list of deliverables", and "Allows attachments or links to specs" were ranked close together.

From the Managers perspective, three criteria were nearly tied for first place: "Creates issues/questions list for customer to review", "Tracks name/date/time of approvals", and "Allows different approvers for different spec types".

From the IT perspective, "Easy to maintain" and "Can be used by other sites" were ranked the highest.

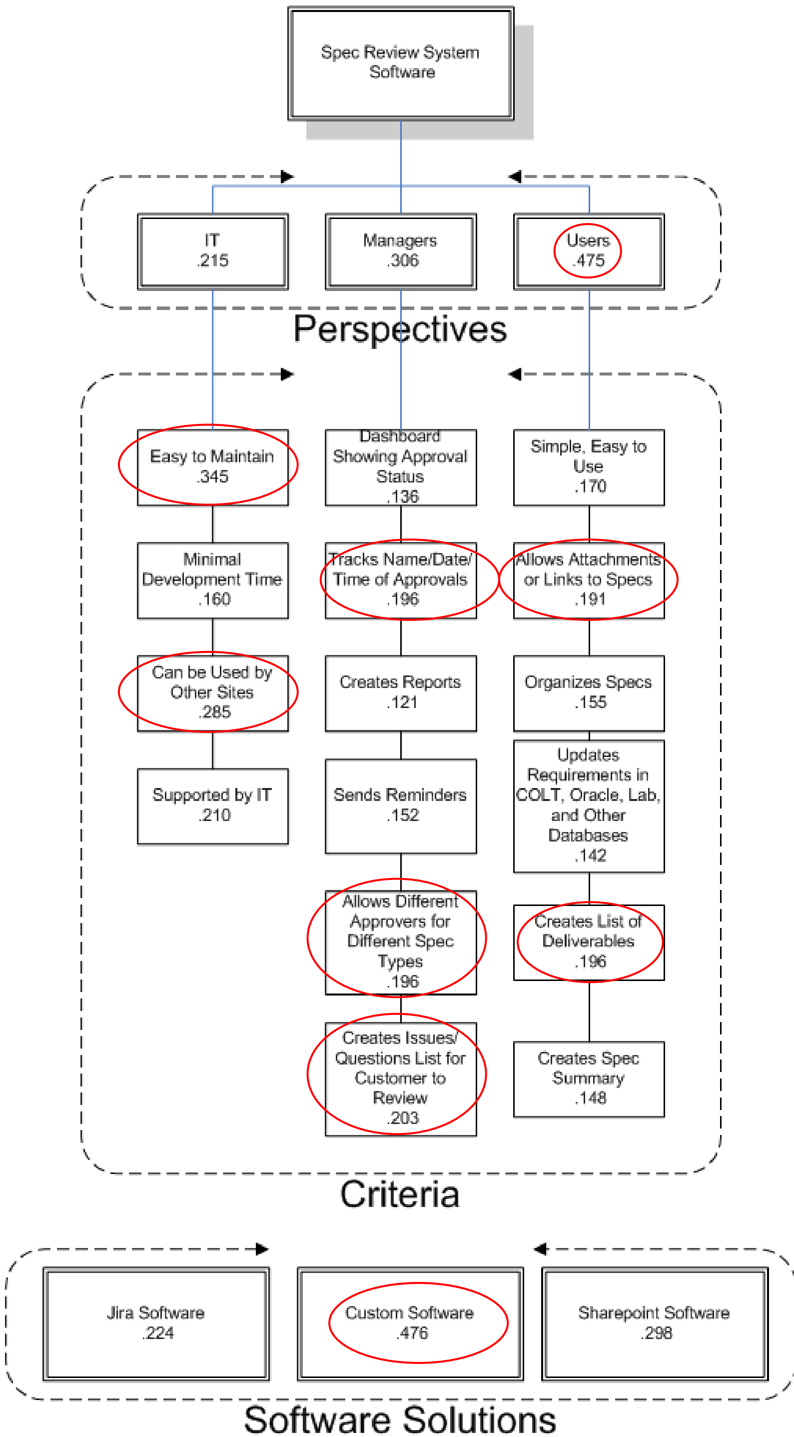

Figure 10. Perspectives and decision criteria weights

This HDM analysis suggested that it could do more than confirm management's initial impression of the best approach, and that it could also provide a ranked order set of criteria that could be used to specify the design of this custom software. Since the criteria are grouped under each perspective (IT, Management, and User), this ranking is not immediately obvious from the individual results. However, a “pseudo ranking” could easily be obtained by normalizing the results from each perspective (see Figure 10). For example, if an expert ranked each of the four IT criteria as 0.25, these scores could be normalized over the total of sixteen criteria to give a ranking of 0.0625:

$$0.25*4/16 = 0.0625$$

The compiled results for this approach to normalizing the data are shown in Figure 11.

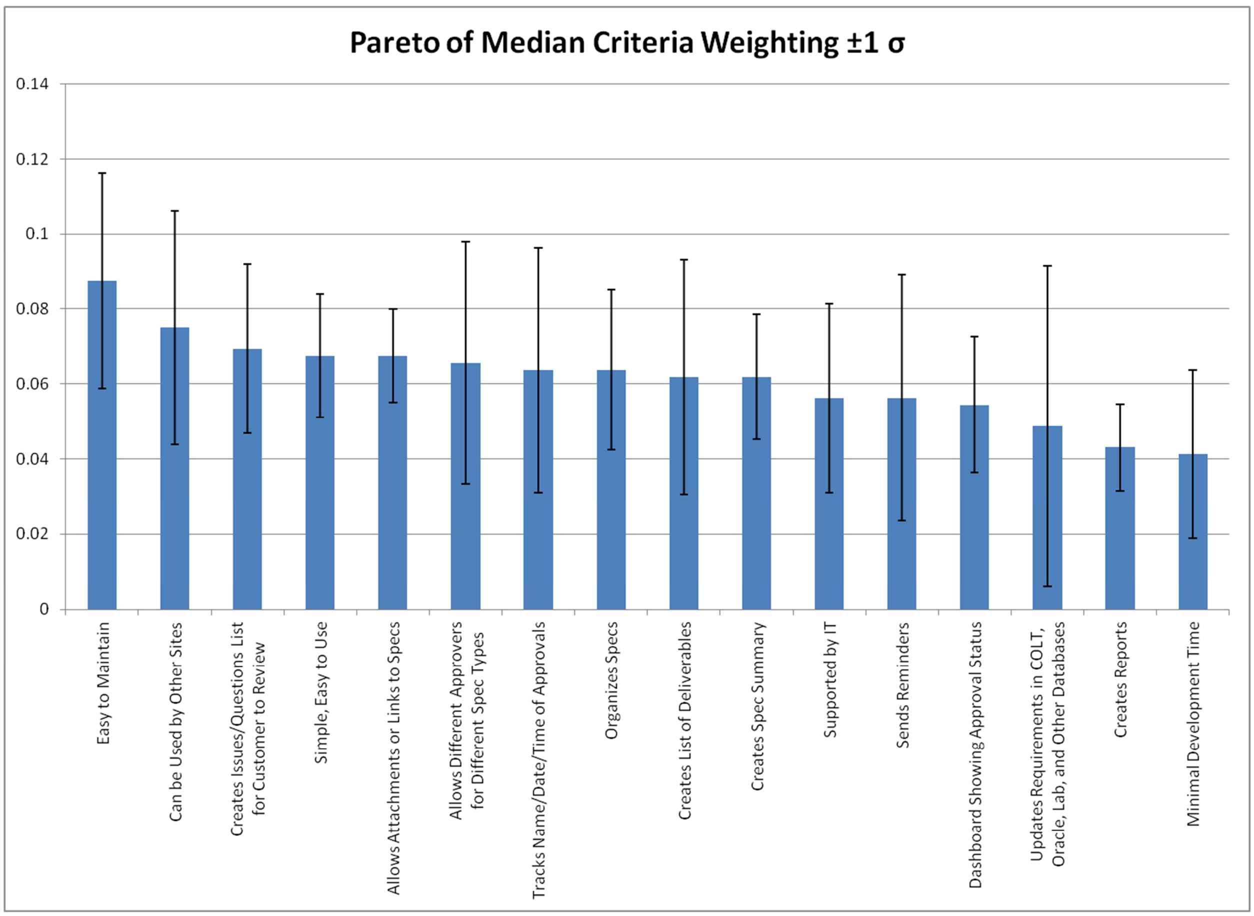

Figure 11. Pareto of criteria weighting (median ±1 σ)

Normalizing the data in this way has an obvious flaw, since this HDM model was not designed to solicit pairwise comparisons between each criteria, but only between the criteria within each perspective. Nevertheless, it provided a way to look at the data that generated discussion and could be useful for eliminating some criteria, potentially saving time and frustration.

Because of the variation in expertise between the respondents, which implied the potential for outliers, I chose to present the median values in this Pareto rather than the means, although as can be seen from Figure 12, the difference between the medians and means was not great. By arranging them in four groups (high, medium-high, medium-low, and low), it appeared that certain criteria were deemed by many as relatively unimportant (#13 – #16) and that these might be considered dispensable criteria. For example, a dashboard showing approval status, the ability for the software to update requirements in other databases, the ability for the software to create reports, and minimal time to develop the software was not viewed by most as critically important. The next level (#9 – 12) might be considered "nice-to-haves" rather than critical requirements.

On the other hand, experts considered that the software did have to be easy to maintain, useful to other sites, capable of creating a customer issues list, and be easy to use (#1 – #4). Other criteria that were ranked lower but had a relatively small standard deviation, such as allowing the addition of attachments or links to specs, allowing different approvers for different spec types, tracking names and dates of approvals, and spec organization, could reasonably be added to the requirements for the custom software given overall agreement in the ranking (#5 – #8).

Criteria	Median	Mean	StDev
1. Easy to Maintain	0.088	0.086	0.029
2. Can be Used by Other Sites	0.075	0.071	0.031
3. Creates Issues/Questions List for Customer to Review	0.069	0.076	0.022
4. Simple, Easy to Use	0.068	0.064	0.016
5. Allows Attachments or Links to Specs	0.068	0.072	0.012
6. Allows Different Approvers for Different Spec Types	0.066	0.074	0.032
7. Tracks Name/Date/Time of Approvals	0.064	0.074	0.033
8. Organizes Specs	0.064	0.058	0.021
9. Creates List of Deliverables	0.062	0.074	0.031
10. Creates Spec Summary	0.062	0.056	0.017
11. Supported by IT	0.056	0.053	0.025
12. Sends Reminders	0.056	0.057	0.033
13. Dashboard Showing Approval Status	0.054	0.051	0.018
14. Updates Requirements in COLT, Oracle, Lab, and Other Databases	0.049	0.053	0.043
15. Creates Reports	0.043	0.045	0.012
16. Minimal Development Time	0.041	0.040	0.022

Figure 12. List of criteria weighting (median, mean, and standard deviation)

Reviewing the Results

These results were met with some disappointment (and a little sarcasm) by management, who felt that some of the experts missed the point of the exercise. From his perspective, regardless of the ranked results, the software had to be capable of understanding the status of submittals (i.e., include a dashboard), and be capable of sending reminders, neither of which had made the top criteria ranking. He believed this was due to the mixture of expert perspectives, particularly those who would not have to use the software once it was developed (i.e., the IT group).

It is arguable that these criteria are not actually needed, since a smoothly functioning spec review system may not require nagging emails or a management dashboard. Nevertheless, it is clear from this analysis that even with the ranked ordering of criteria, some management judgment in the final criteria selection will be required.

Limitations

Although all analytic hierarchical process approaches to decision making, including the HDM approach, are useful for making a complex decision with hard-to-quantify criteria more tractable, this approach also has several obvious limitations: it is time-consuming; it requires clear communication of the meanings of each perspective, criteria, and option to ensure the expert opinions are consistent; and the quality of the results depends on the depth and breadth of knowledge of these experts.

FUTURE RESEARCH

The primary question addressed in this paper was, “What is the best software solution for a spec review process?”, but a deeper question is whether it is possible to use an HDM approach to help develop that software by prioritizing the design criteria.

New HDM Decision Questions

Future research might consider the following questions:

- Was it a coincidence that both approaches yielded the same overall result? Or did a "fast and frugal" approach effectively arrive at the same high level decision of which software option to choose? What would a comparison of both approaches (intuitive and analytical) with each expert show, and would it generate a better overall solution?
- Is it possible to use a design of experiments (DOE) approach to estimate the results of an exhaustive pairwise comparison of all criteria in order to create a valid ranking of the criteria and simultaneously reduce the time involved? Perhaps a "screening" DOE could be used to weed out less important criteria first?
- What other hardware or software could be designed using the HDM approach to criteria selection?

HDM Process Improvements

The HDM tool is somewhat unforgiving, and I discovered the hard way when I accidentally clicked the "Delete this" hyperlink next to one individual's analysis results that the software does not give one a second chance before deleting. Also, the software does not provide the average ranking for each criteria for the aggregated data, only for each individual respondent. Perhaps the most important limitation of the tool, and this is inherent in the approach, is that by apportioning 100 points between two criteria, the ranking is non-linear. That is, if I apportion 75 points to one option, it may "feel" as if I'm choosing to rank my favored choice as slightly better than the alternative, since I've only moved the slider halfway to the right of center. But in fact, I am saying that my favored choice is three times as important as my least favorite of the pair. This method of ranking is not intuitive, and might lead to over-ranking criteria.

REFERENCES

Gigerenzer, G., & Goldstein, D. (1996). Reasoning the fast and frugal way: Models of bounded rationality. *Psychological Review, 103*(4), 650-69.

Gladwell, M. (2005). *Blink : the Power of Thinking Without Thinking*. New York: Little, Brown and Co.

Kahneman, D. (2011). *Thinking, Fast and Slow.* New York: Farrar,Straus and Giroux.

Leigh, A. (2015). How behavioural economics does and can shape public policy. *The Economic and Labour Relations Review*, 26(2), 339-346.

Milkman, K. L., Chugh, D., & Bazerman, M.H. (2009). How Can Decision Making Be Improved? *Perspectives on Psychological Science, 4*(4), 379-383.

Moxley, Jerad H., Ericsson, K. Anders, Charness, Neil, & Krampe, Ralf T. (2012). The Role of Intuition and Deliberative Thinking in Experts' Superior Tactical Decision-Making. *Cognition*, 124(1), 72-78.

Tu, N., Zhang, T., He, Q., Zhang, H. & Li, Y. (2011). Applying combined AHP-QFD method in new product development: A case study in developing new sports earphone. *International Conference on Management Science and Industrial Engineering (MSIE)*, 80-85.

Saaty, T. (2008). Decision making with the analytic hierarchy process. *Int. J. of Services Sciences,1*(1), 83-98.

Salgado, E., Salomon, V., & Mello, C. (2012). Analytic hierarchy prioritisation of new product development activities for electronics manufacturing. *International Journal of Production Research*, 50(17), 4860-4866.

Yuen, K.K.F. (2014). Fuzzy Cognitive Network Process: Comparisons With Fuzzy Analytic Hierarchy Process in New Product Development Strategy. *IEEE Transactions On Fuzzy Systems*, 22(3), 597-610.