Naval Facilities Engineering Command
EXWC Engineering & Expeditionary Warfare Center

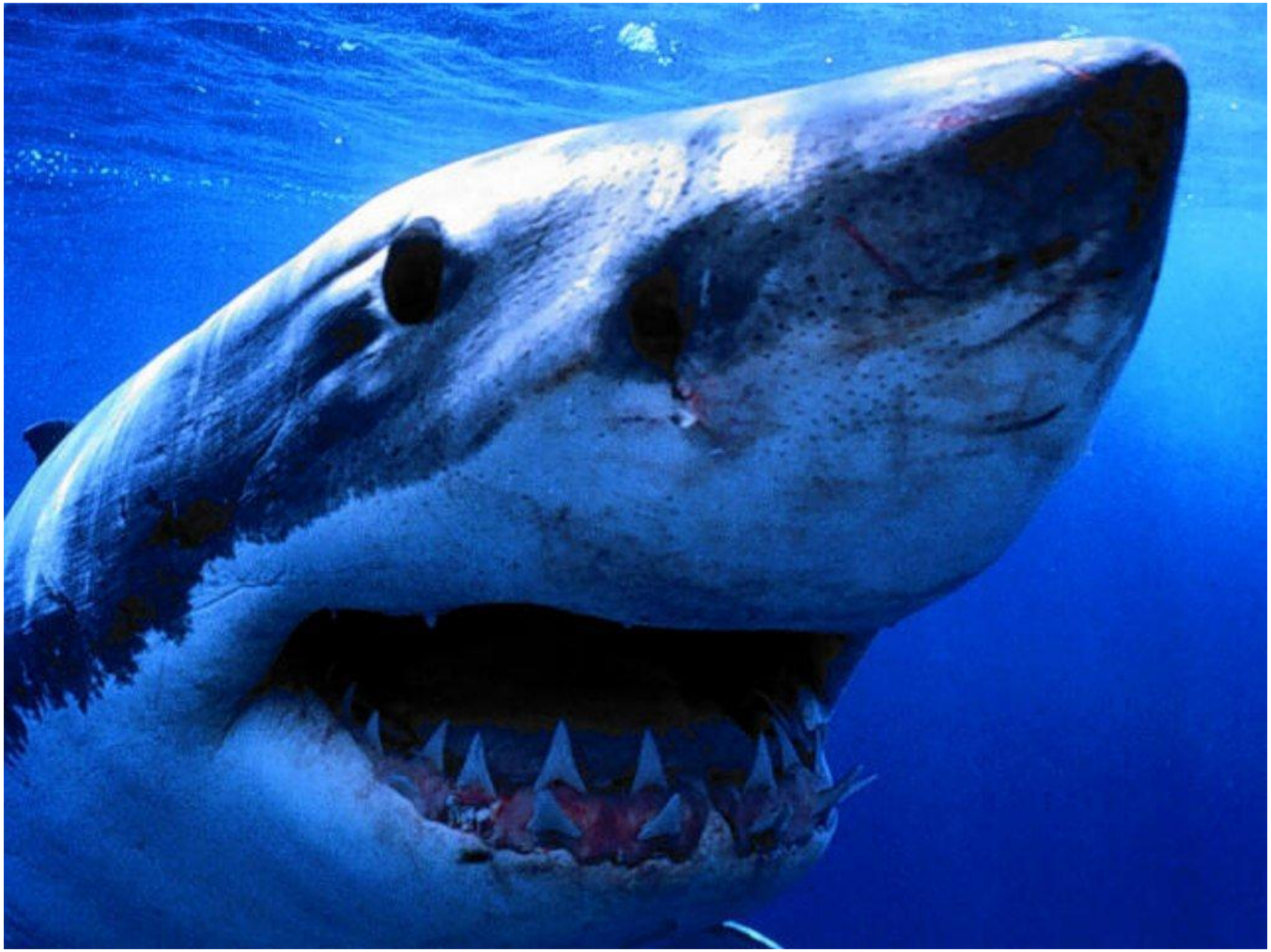
Explosives Safety Siting Quantity-Distance Engine and Flowcharts

Criteria Merge and Software Rewrite

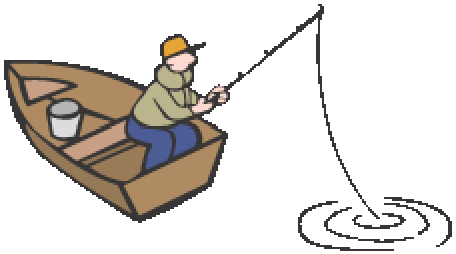
Cameron Stewart
Research Structural Engineer
CI7 Explosion Effects and Consequences Division

Introduction

Introduction (cont)



Introduction (cont)



(Apologies to Steven Spielberg and Peter Benchley)

Manage RISK

Executive Overview

- The Quantity-Distance (QD) Engine for the Explosives Safety Siting (ESS) software is being rewritten to simplify the programming, reduce maintenance costs, and switch the programming language to promote future maintainability and adaptability.

Background—Site Planning

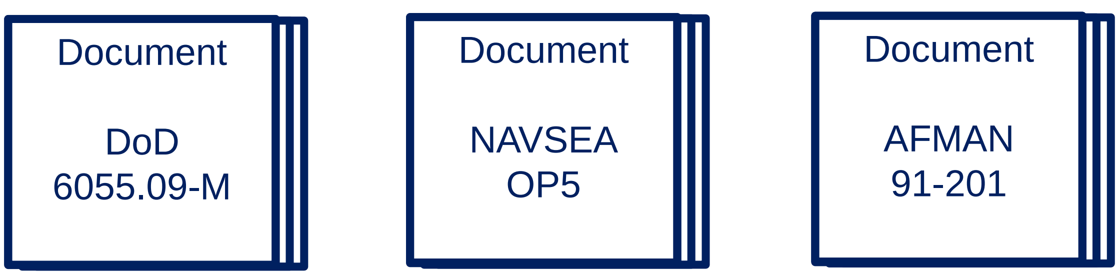
- Explosives safety site plans are required whenever new construction or explosives operations change at any installation within the United States Department of Defense (DoD).

- The complexity, time-investment, and high level of risk involved with this process led to the development and deployment of Explosives Safety Siting—ESS software to automate the majority of it.

Learn more about ESS and its future development during the Thursday afternoon—DDESB - Tools and Practices technical session!

Background—ESS

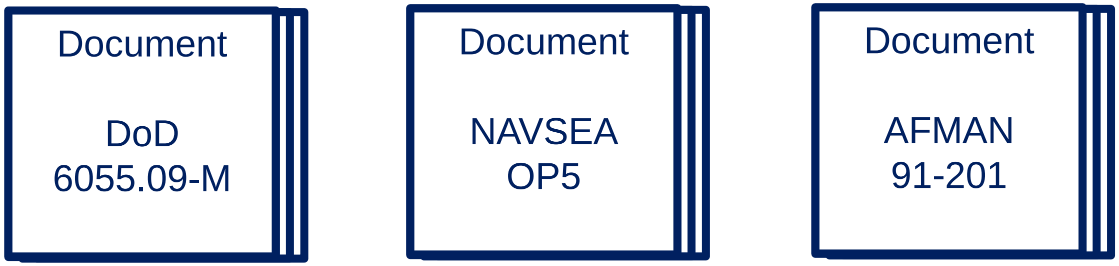
- ESS software uses its Geographic Information System (GIS) platform for much of its functionality
- A key component of ESS has been programmed separately to calculate the Explosives Safety Quantity Distance (ESQD) arcs based on the criteria found in:
  - DoD 6055.09-M (DoD criteria)
  - NAVSEA OP5 (United States Navy criteria)
  - AFMAN 91-201 (United States Air Force criteria).



Background—ESS QD Engine

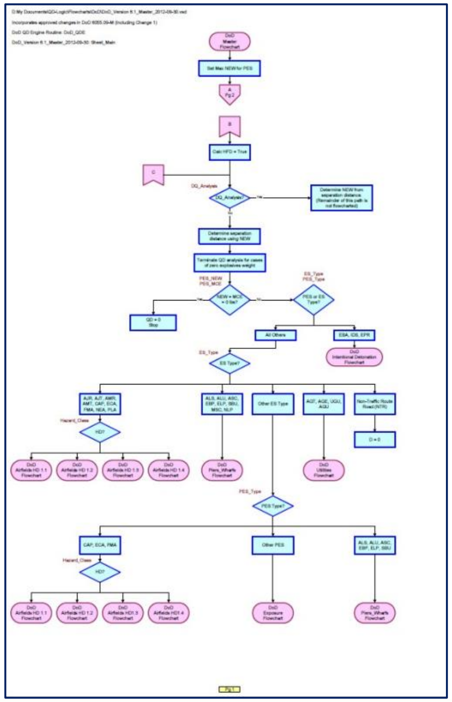
- This separate ESQD arc calculation program is called the QD—for "Quantity-Distance"—Engine and it was programmed in VB.NET.

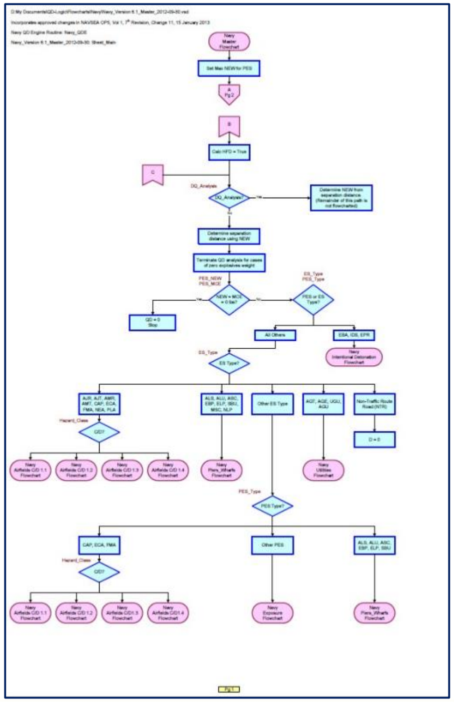
- The QD Engine was actually broken down into three separate engines as three separate software files, one for each criteria manual.
  - The DoD Engine
  - The Navy Engine
  - The Air Force Engine

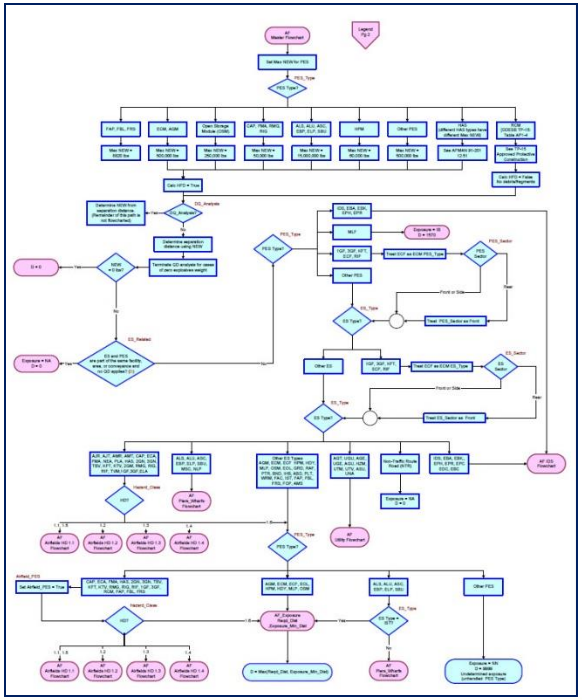


Background—ESS QD Flowcharts

- Three sets of flowcharts, one for each engine, were created in conjunction with these QD Engines as programming guides to demonstrate in each case how the criteria would be used to calculate ESQD arcs for any scenario.







Background—ESQD Calculations

Determine the following:

- PES Type
- ES Type
- Attributes associated with the PES and ES
- Define the explosive associated with the PES

Based on the information in the previous step, determine the Exposure:

- Inhabited Building Distance—IBD (most conservative)
- Public Traffic Route Distance—PTRD
- Intraline Distance—ILD
- Intermagazine Distance—IMD (least conservative)

Based on the assigned exposure as well as the amount and classification of the explosive associated with the PES, the ESQD is determined.

Most often this takes the form of the equation:

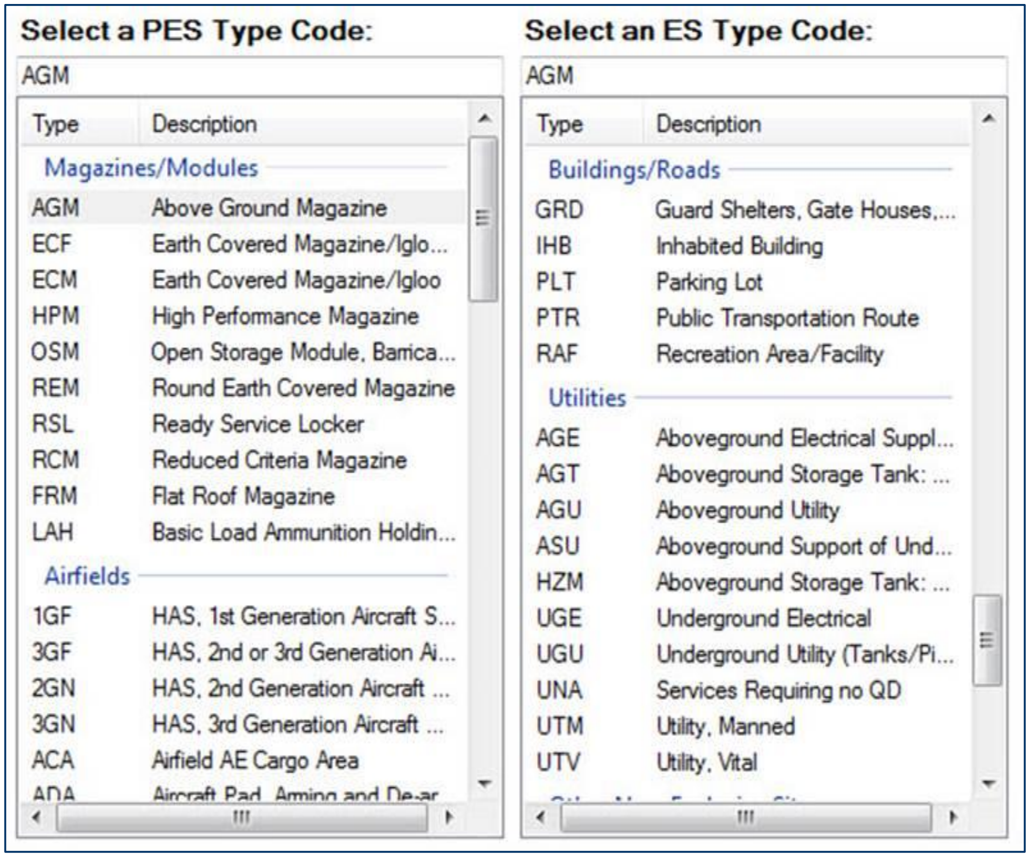
$$ESQD = K*(NEW)^{1/3}$$

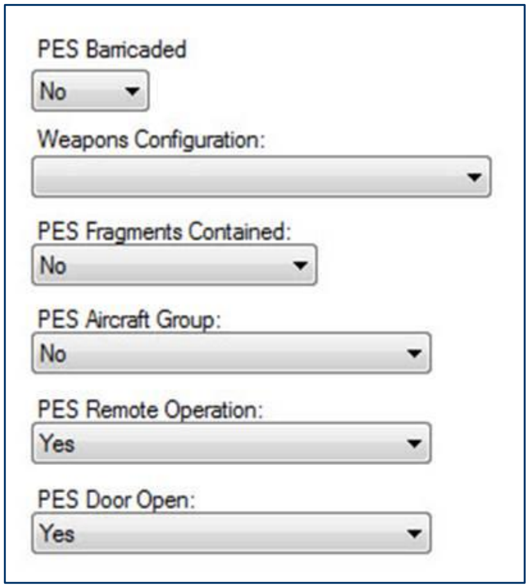
QD Engine—Support Programs

- **The QD Calculator application allows users to perform ESQD calculations outside of ESS**
- **To ensure the QD Engine gives accurate results, a library of validation problems was created in MS Access called the Validation Database**
  - **Each problem was created separately from the QD Engine using the flowcharts and criteria manuals**
  - **Problems were created for each version of the QD Engine (DoD, Navy, and Air Force)**
  - **Currently there are 6,611 unique problems in the library**
  - **A separate program was developed to automatically test the QD Engine using the Validation Database**
    - The results from the QD Engine are compared with those entered into the database

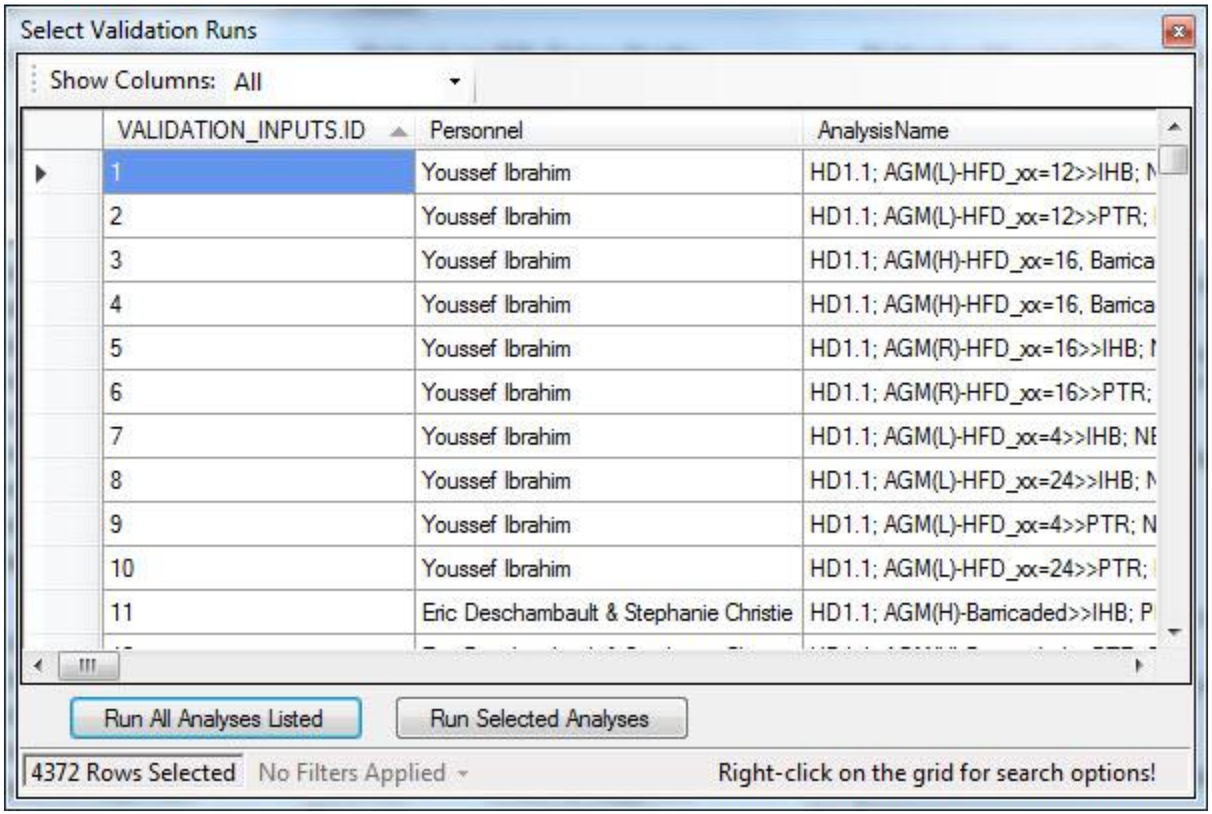
Background—QD Calculator

- **Screen shot examples of input screens for the QD Calculator**

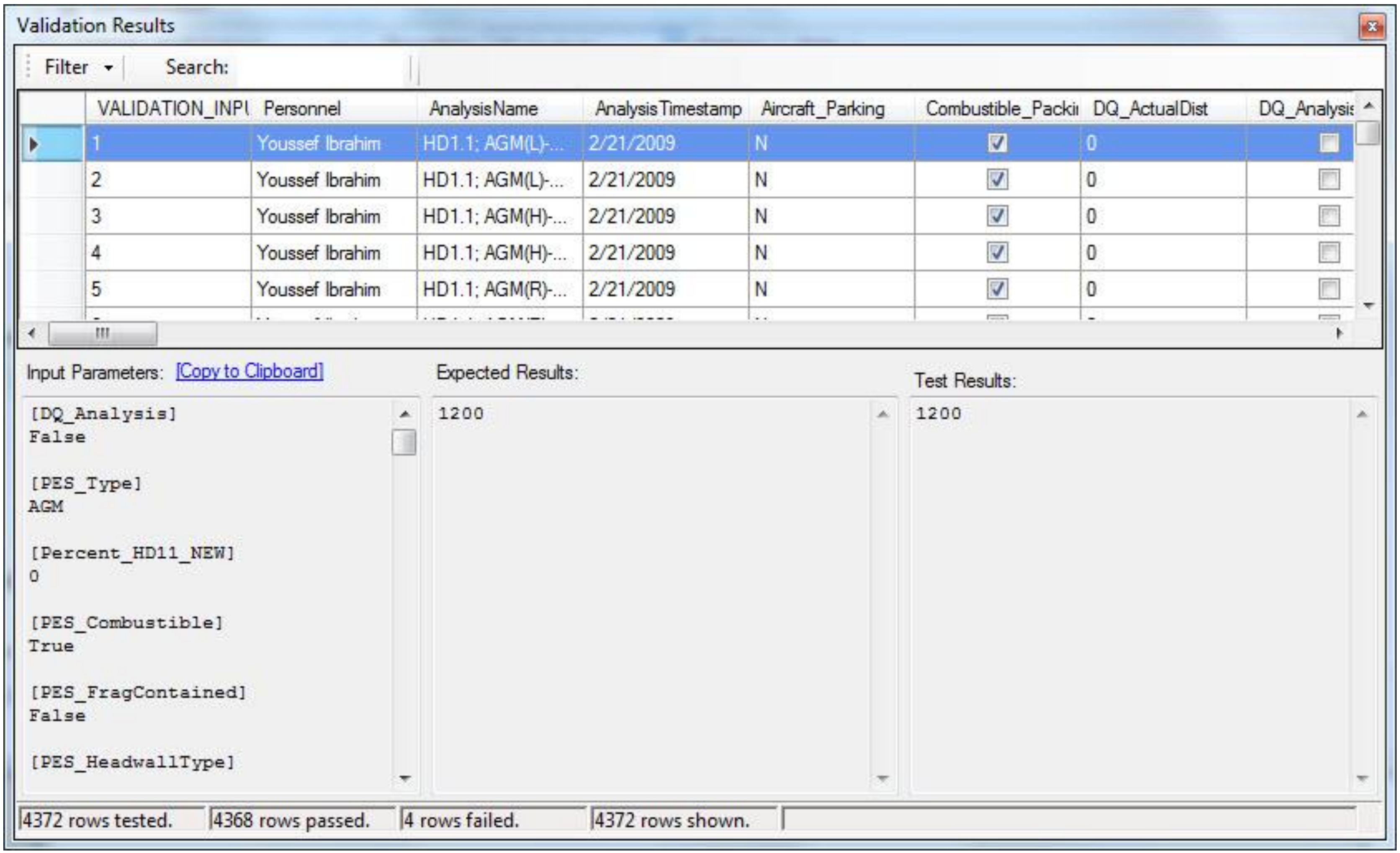




QD Engine—Validation Problems



QD Engine—Validation Results



Flowchart and QD Engine Merge

- Different entities developed and maintained the separate QD Engines and associated flowcharts.
- This has resulted in unnecessary redundancies in the flowcharts and QD Engine and added complexity whenever updates to the criteria necessitated updates to the flowcharts and QD Engine.
- In order to promote the long-term sustainability of ESS, it was decided to merge all of the flowcharts and engines.

QD Engine Rewrite

- ESS web-based application currently under development.
- Interest has been expressed to develop a mobile application of the QD Calculator in the near future.
- VB.NET is incompatible with web and mobile programming.
- VB.NET is falling out of favor as a programming language.
- C# can be used in both web and mobile programming.
- C# is widely used and well-supported.
- C# requires a relatively small learning curve for EXWC personnel.
- EXWC recommended and obtained approval to rewrite the QD Engine in C# as part of the QD Engine merge project.

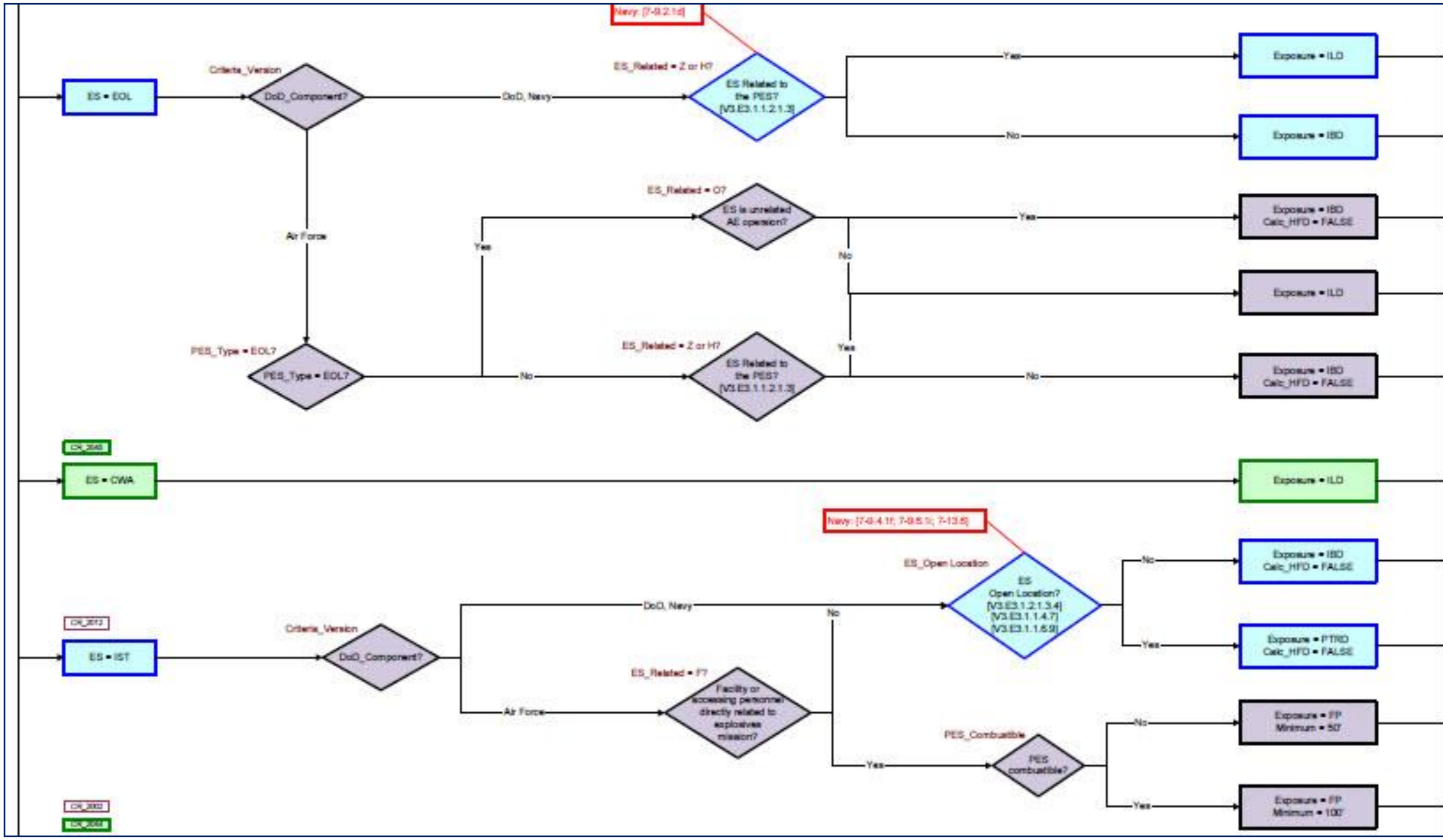
Merge Process—Project Scope

- There were only minor differences between the criteria associated with both the DoD and Navy
  - EXWC had developed and maintained both the DoD and Navy engines and flowcharts.
  - The differences between the associated engines and flowcharts were minor.
  - The level of effort required to merge the Navy with the DoD was minor.
- Having a different author behind the Air Force Engine and flowcharts led to a great many differences between that engine and flowchart set and those developed by EXWC.
  - EXWC would have to make major changes to ensure the logic flow of the Air Force criteria in the merged engine and flowcharts matched that used for the DoD and Navy criteria.

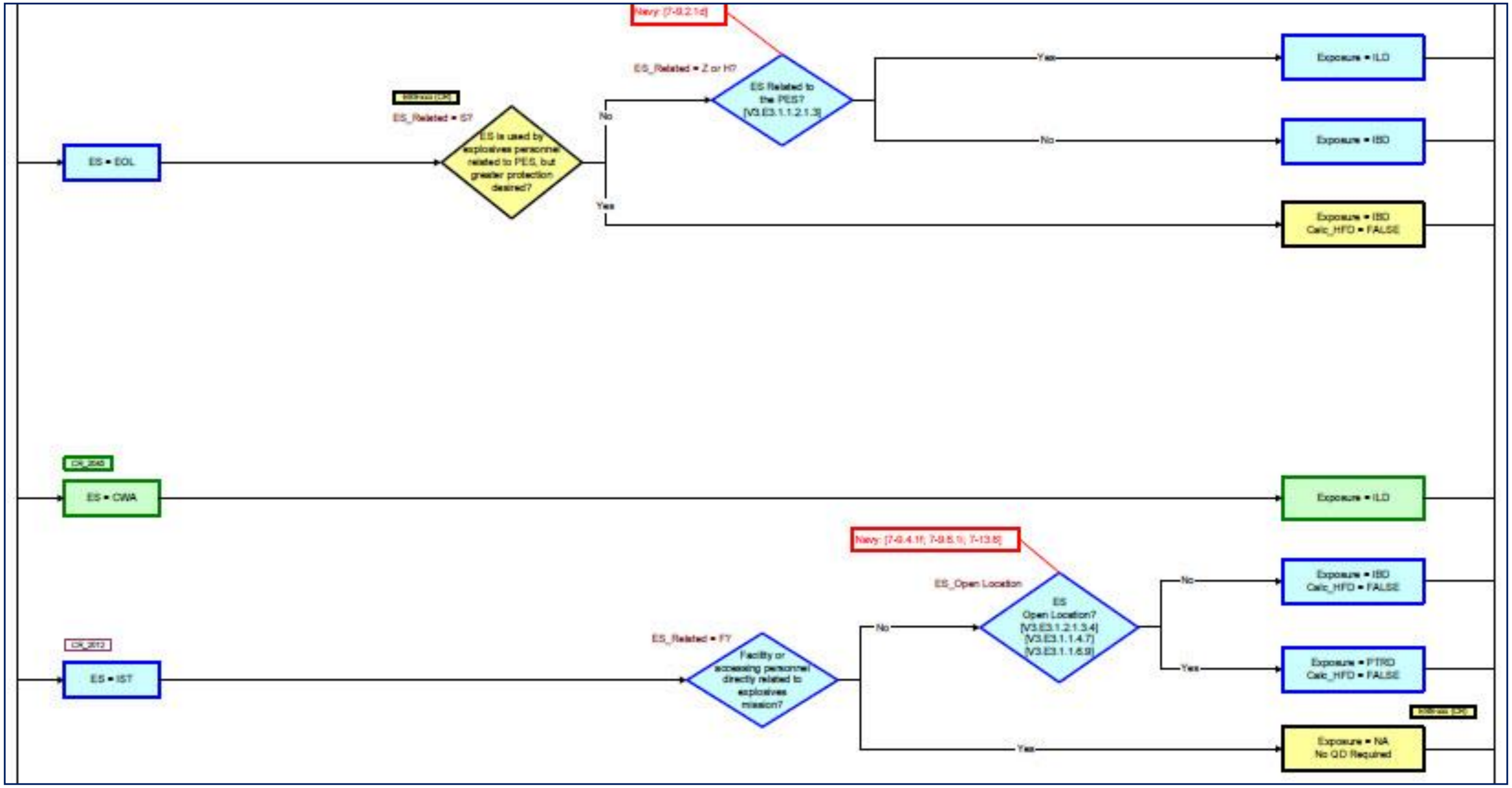
Merge Process—Criteria Review

- Early in the merge process EXWC prepared very rough drafts of the merged flowcharts.
- A meeting was held with representatives from the DDESB and explosive safety criteria experts from each service to review the flowcharts.
  - There were many more similarities between the different criteria manuals than had been thought previously.
  - The merged flowcharts could be greatly simplified (see the next slide for an example).
- An iterative process followed where changes were made to the flowcharts, which were reviewed by the same group of experts, and needed changes were then made and reviewed.

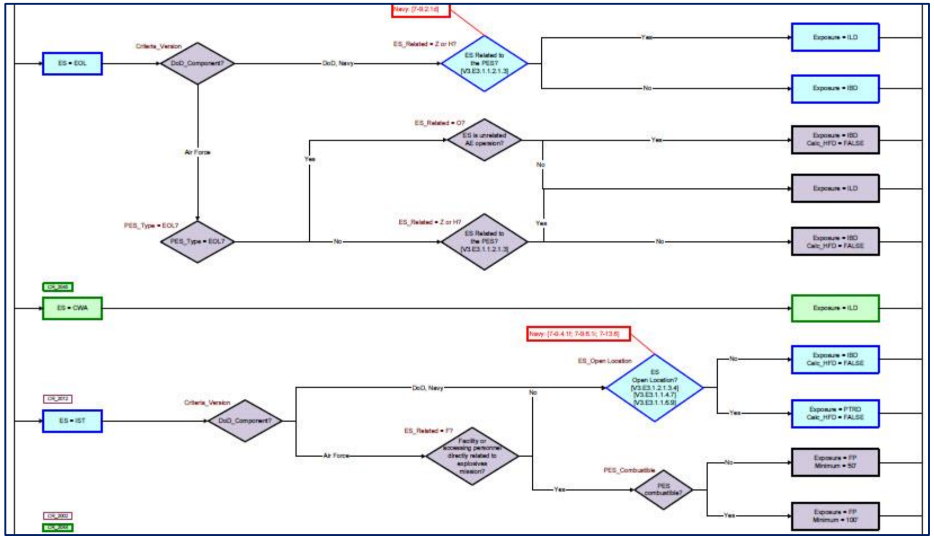
Merge Process—Rough Draft Merged Flowchart

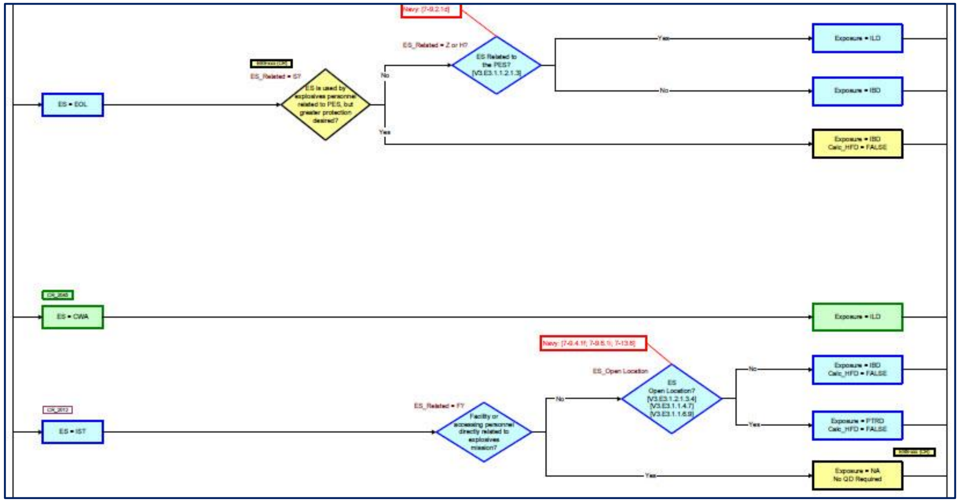


Merge Process—Revised Merged Flowchart



- **Side-by-side comparison:**

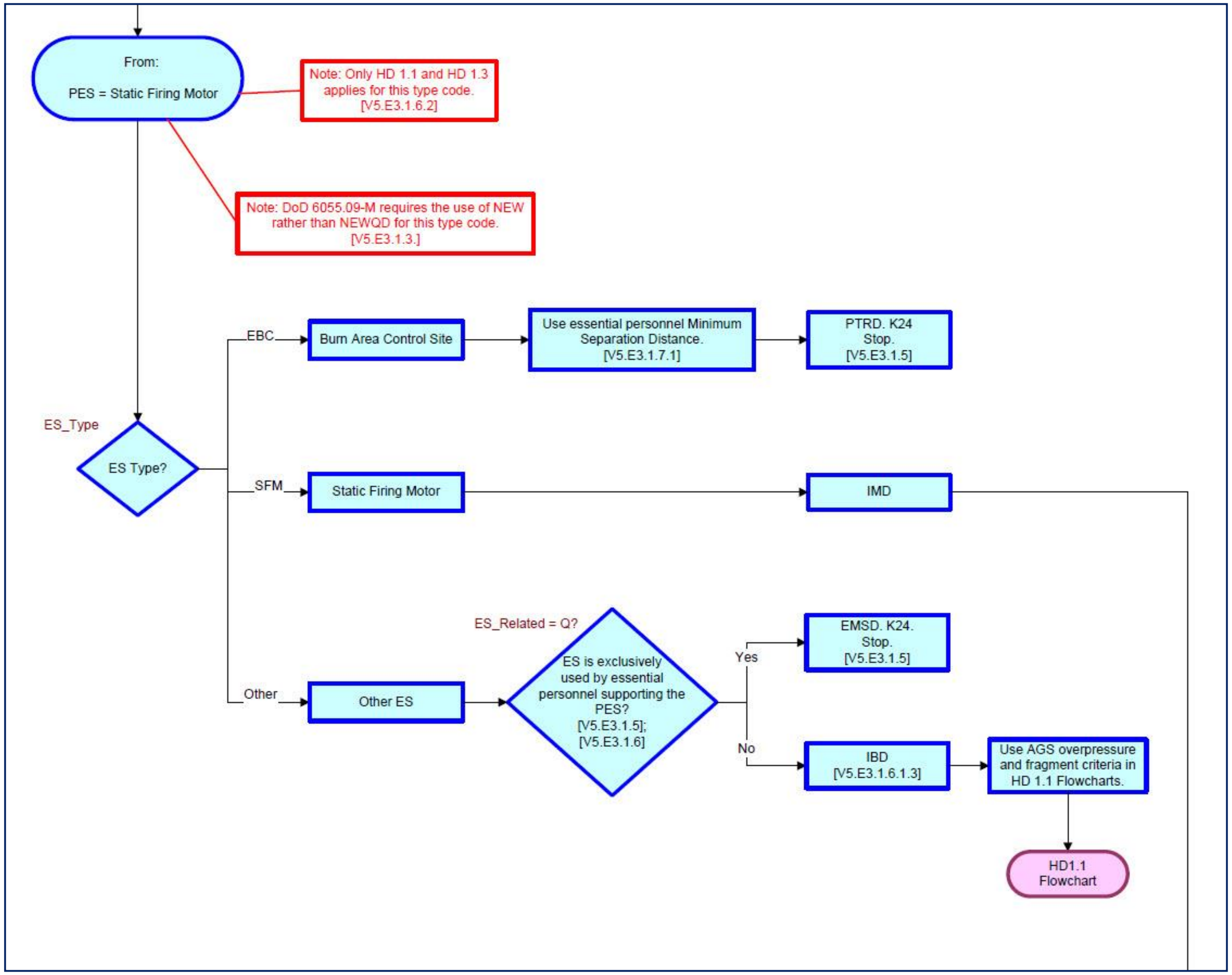




Merge Process—Inclusion of New Criteria

- Another aspect of the flowchart revision was the inclusion of new criteria which had been approved but not yet published in any of the manuals nor included in ESS.
  - Most of these new additions pertained to the criteria governing intentional detonations and utilities.
- The flowcharting logic for each of these additions had to be drafted and then subjected to the same process of review and revision as with the merging of the criteria.

Merge Process—New Criteria Example



Merge Process—Results

- By merging the flowcharts from three separate sets totaling 276 pages down to one common set totaling 152.
- Less time and effort will need to be spent in maintaining them in the future.
  - Instead of checking three different sets of flowcharts to review and revise criteria, only one set, and often only one page needs to be checked.
  - Often the criteria between all three manuals is the same.
- By merging the flowcharts first, a blueprint was created for merging the QD Engine, making that process more efficient.

QD Engine Merge and Rewrite

- The process to rewrite the QD Engine was broken down into stages in order to make the transition from the multiple engines in VB.NET to one engine in C# as smooth as possible.
  - Stage 1 was a straight conversion of the existing VB.NET QD Engine into C#.
  - Stage 2 used the DoD Engine as a starting point to develop one combined QD Engine with all of the criteria contained in the previous 3 engines incorporated in one engine.
  - Stage 3 involves incorporating the Validation Database with the new QD Engine, adding to and revising the validation problems as needed, then testing and debugging the QD Engine.
  - Stage 4 involves any additional changes to the structure of the QD Engine and its supporting files to enable seamless integration with ESS in both its desktop and web-application platforms.

QD Engine Merge and Rewrite—Stage 1

- Stage 1 involved using an automatic tool to convert the VB.NET code into C# and then correcting any remaining bugs preventing the existing, unmerged engines from running without errors.
- No changes were made to the way the engines calculated criteria, and nothing was done further at this stage towards merging the engines.
- This allowed EXWC to use the Validation Database to validate the C# conversion.
  - This ensured that the new programming language would work and that the merging and revisions necessary would be based off a stable program.
- At the conclusion of this stage, the entire VB.NET code for the QD Engine was in C# and the results exactly matched those obtained from the VB.NET version.

QD Engine Merge and Rewrite—Stage 2 Overview

- Stage 2 involved the actual merging and rewriting of the 3 engines into one common engine.
- The C# version of the DoD Engine was used as the starting point.
- Two programmers went through each portion of the code that related to the sections in the merged flowcharts and made changes as necessary.
  - The intent was for the logic in the code to match the logic in the flowcharts.

QD Engine Merge and Rewrite—Stage 2 Process

- Much effort was made to reuse existing code in order to expedite the revision process.
  - Complete rewrites of portions of the code were done when it was possible to use more efficient programming techniques than were used previously.
    - Such revisions only made sense if the techniques were not too advanced.
      - EXWC programming policy has been that internally developed and maintained programs could be understood by those with basic training in programming but did not hold degrees in computer science and engineering.
      - This also ensures that programming does not get so specialized that turnover in project personnel jeopardizes EXWC’s ability to support its software.
  - The inclusion of new criteria also necessitated writing new code.
- Limited testing has been done to ensure at least partial functionality of the rewritten and new portions of the code.

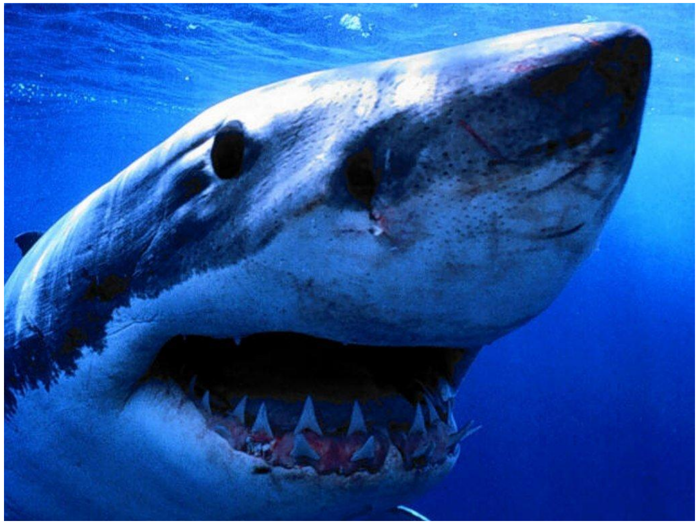
QD Engine Merge and Rewrite—Stage 3

- Stage 3 will involve reconfiguring the programming related to the use of the Validation Database so that it can be used to test the new QD Engine
- Revisions to existing problems and the addition of new problems in the Validation Database will be made.
  - This will account for changes in criteria and ensure that each branch in the merged flowcharts has a corresponding validation problem.
- The updated Validation Database will be used to debug the new QD Engine.
  - Any remaining bugs not found using this library should be minimal.
- At the conclusion of this process, the new QD Engine can be used in ESS and with a high degree of confidence that results will be accurate.

QD Engine Merge and Rewrite—Stage 4

- The last stage will be to make any additional changes to the structure of the QD Engine and its supporting files to enable seamless integration with ESS.
  - In the future ESS will be in both a desktop and web-based format.
  - The new QD Engine will need to function in both formats.
- This will require coordination between the general ESS development team and the QD Engine development team.

Review—Managing Risk!



Conclusion

- Merging the flowcharts and rewriting the QD Engine has been a significant task which will pay dividends in the future.
  - The cost of maintenance and risk of error have both been reduced.
  - Explosives safety siting criteria has been simplified as the review process worked to minimize the differences between each organization.
  - The C# version of the program has become more efficient and is positioned for continued relevance as a commonly-used, well supported language adaptable to different applications.
- The new QD Engine and flowcharts are well positioned for sustainable maintainability and adaptable to the future development of ESS as it evolves to better serve the warfighter.

Questions?

Learn more about ESS and its future development during the Thursday afternoon—DDESB - Tools and Practices technical session!