Attention Naval Architects:
What is your in-house code costing you?

An overview for potential software users and managers

Naval architects, do you remember the slide rule and the abacus? Hopefully not, but you may still be using the 20th century equivalent depending on your design software! In the marine industry, naval architects and engineers rely on software – both home-brewed and commercial – to provide real-world solutions for transporting cargo, ferrying passengers, and meeting other mission criteria. As a highly specialized and narrow segment of the overall transportation industry, the first computer programs were developed by the naval architects and engineers who required them. Each company had its own codes, spreadsheets, and in-house software with its own unique tweaks. In the past it was necessary, but it has never been ideal.

Narrowing the focus to hydrodynamics and propulsion system simulation, many software codes are based on published empirical formulas. These mathematical relationships between vessel parameters and vessel performance date back to the early 1960's (as do some of the in-house codes in circulation today).

What is my in-house code costing me?

For naval architects, accurate predictions have become a staple of customer expectations. Your reputation is directly tied to the performance of your software. Your software is critical – it should always be an asset, not a liability. HydroComp is dedicated to providing software for engineers and naval architects – as a result, we frequently encounter companies that are deeply invested in their own in-house developed tools. In fact, HydroComp's niche market means our biggest competitive product is a customer's own in-house codes! If you are using in-house tools, you must ask yourself – what is my in-house code costing me?

… But my code is "free"!

The first issue with in-house codes is the illusion that the program is "free". Certainly, some codes are freely available, and an engineer may leverage a free code as a starting point. Monetarily, yes the code is free, but the reality is that the true costs for deployment, development, maintenance, and training of the in-house code is significantly larger than most companies realize. Development and maintenance are required in small pieces, the costs are difficult to trace, and full time development difficult to justify.

… But we can code!

Engineers are often capable of writing their own tools – spreadsheets, command-line programs, or small batch scripts, to name a few variants. There is a natural desire to understand, develop, and "own" the tool or solution. It is important to remember that an in-house code is still an investment of your time and energy. In-house codes require perpetual development. The reality is that technological advances tend to outpace an individual programmer's ability to implement and validate new features and analyses (e.g., emission requirements, IMO standards, propeller class thickness rules, new prediction methodologies for resistance and propulsion).

So ask yourself – can you dedicate the time to develop codes? As a business, your core competency is not software development. In-house codes are rarely shared, almost never sold to competitors. This means that every hour spent developing software is costing your business. Each development hour is one less hour devoted to increasing financial revenue for the company.

… But our guy understands!

Most software can be used without understanding the underlying code. Contrast this with software developers, who have an intimate understanding of the internal codes. The problem with developing your in-house code is that these mathematical underpinnings and program structures may only be truly understood by a single individual (and this person may not even be within your organization!). Transferring this knowledge is incredibly difficult, and this presents a very real risk of "knowledge loss" for the organization, should said individual leave. To keep abreast of the industry, any software will need periodic updates. Will this individual be there in the future when your code needs an update?

... But our code is simple!

Each in-house code is a unique application – even the simple codes are a "black box" to new employees and also to your customers. The truth is, most people have not seen or used any form of your in-house code. Non-standard applications present a barrier to adoption and use by new employees. With few employees knowledgeable about a critical code, an engineering bottleneck can often form, leading to backlogs of work for knowledgeable employees.

Additionally, the next generation of engineers is accustomed to standardized tools with modern graphical interfaces. Likewise, they are use to cross-software integration, where data can be seamlessly transferred between their tools. Many in-house codes were created decades ago and have only a command-line interface. Data-transfer is performed manually and is prone to entry mistakes.

All software has a shelf life, especially in a world where technology develops exponentially. No matter how simple your code may be, when it is "long-in-the-tooth" there is always the risk that support for specific hardware or software will end and it will just stop working. HydroComp often encounters companies running Windows 95 and XP systems in order to keep legacy software working.

OK, tell me about commercial codes…

Commercial software has several advantages over in-house developed codes. The number-one reason is accuracy. A dedicated software company has a core-competency in… well, developing software! HydroComp, in particular, exclusively develops software for naval architects and marine engineers. Written by a team of naval architects, engineers, and software developers (and working closely with reputable professionals from industry), our team is 100% focused on the accuracy of our codes. We are extremely strict with prediction methodologies and validation studies. Just because a method is published in a scientific journal does not mean it is accurate enough for our software or for our customers.

A large user-base is also a benefit of commercial software, as it leads to decreased liability and risk. It allows development cost to be spread amongst many similar companies, allowing individuals to benefit from new features. Commercial software is extensively validated and actively used by large numbers of people, which in most cases leads to a huge reduction in bugs and increase in analytical accuracy.

Another key advantage of commercial software is usability. Nearly any engineer can take a published formula and create a spreadsheet calculation, but a major problem is the scope of the formula. To make these methods reliably accurate, commercial software such as NavCad or PropElements will review the accuracy of the prediction for the different input parameters and determine reasonable limits and constraints. This ensures the end-user will always get an appropriate answer within the range of the original data – extremely important when your reputation is on the line!

The other important aspect of usability is the program interface. Contemporary software, of course, is based on graphical user interfaces (GUI), not command-line interfaces. Why delay switching, especially where new engineering talent is accustomed to full-featured GUI's? Commercial software uses a standardized interface for all users and is compatible with modern platforms (such as Windows 10). This leads to the development of transferable skills – meaning you can hire an engineer who is already familiar with the tool. This provides flexibility in an organization when an employee transfers or staff changes, mitigating the risk of "brain drain" for the organization.

Commercial software companies also provide support. The dedicated development team at HydroComp provides extensive software documentation, telephone and email support for questions, and software training courses for new and advanced users. This level of assistance is impossible with in-house codes. Often, training and knowledge can only be passed down by a higher-ranking individual. With commercial software, you have a dedicated external partner to rely upon for software development and technical support.

Growing pains

The transition from in-house capabilities to commercial software is an important decision that more and more companies are facing each year. Change is never easy, but it is a fundamental reality of technology and business. Advances in the marine industry will outpace in-house codes (and likely already have). At some point, you will require capabilities your in-house code does not provide. How do you obtain or develop these capabilities?

Commercial software has many advantages over legacy in-house codes. Decision-makers should understand the above pitfalls with in-house codes and the value of moving to standardized, commercial software. Migration is filled with technical and political obstacles. Companies, their employees, and in particular, the creators of the in-house codes are often "invested" in their tool and resist moving to another solution. Migration is change, and change requires effort, but the key is that this effort to move to a new tool yields a net benefit with software maintenance, knowledge retention, professional training, and a better analytical process that considers more of the problem, ensures higher fidelity solutions, and interfaces seamlessly with other programs.

Keys to success

- Evaluate in-house and commercial codes in terms of mid-and long-term objectives

- Identify key motives for migration and quantify meaningful metrics of success

- Consider potential value but also migration obstacles – risk vs. reward

- Collaborating with a commercial software provider can often solve migration problems in a very efficient manner

Conclusions

In-house codes are adopted largely due to the perceived cost savings. The reality is that in-house codes cost more in the long run in terms of time, resources, and risk. They require resources to develop and maintain, are often a barrier to new users, and lead to a small number of individuals with intimate knowledge about the use and inner working of the code (which presents a risk to the company if this knowledge is lost).

Commercial software has many advantages over in-house codes. These advantages include higher accuracy and reliability, ease of use via a standardized interface, extensive documentation, support, and training materials, and a focused team to continuously improve the software. While the acquisition costs may be higher, there is significant savings over time due to decreased risk for your projects, transferable skill sets for staff, modern user-interfaces, and the most up-to-date prediction methodologies. Remember, doing

things cheaply can ultimately be costly. You must truly ask yourself *"What is my in-house code costing me?"*

2018/10