Q&A: Weighing the Benefits and Costs of Code Reuse

National Instruments Business and Technology Fellow Mike Santori discusses the benefits of code reuse as well as the factors developers need to take into consideration when deciding to integrate third-party IP into their projects.

Q: Other than time savings, what are the benefits of code reuse?

Code reuse is often seen as a method to save development time by cutting project costs and shortening time to market, but it has several benefits besides saving time. Code reuse fundamentally allows algorithmic-level details to be encapsulated, making it easier to understand code by presenting it as high-level functional blocks that can be shared across different teams in an organization. Encapsulating the algorithm allows developers to work at a higher level of abstraction so that they can be more productive.

Another advantage of a shared code base is that it allows multiple developers to integrate IP into their code using similar styles and programming conventions. This permits easy information sharing and allocation of development resources throughout a large organization as well as a smaller overall code base that needs to be maintained.

Lastly, when in the testing and validation phase of a design, it is beneficial for code to be validated as functional modules. Validating the behavior of each block of code dictates known behavior, whether used multiple times in a single project or in many projects. Fixing shared code propagates changes to all current and future designs.

Being able to reuse code makes it possible for teams to make better use of their development resources and produce more maintainable, higher-quality code, all while having the potential to save money and reduce time to market.

Q: Standardization of IP has not happened. Do you believe it will?

The benefits of IP standardization to the end user are tremendous. By having known interfaces to common IP sets, developers can pick IP from the vendor that provides the ideal solution to their problem – whether it is code size, performance or power considerations – with minimal redesign of existing architectures. The intention of developing IP is to maximize the end user’s productivity, and a standard set of interfaces and usage models would go a long way toward achieving that.

Because of the benefits, IP standardization is likely to occur through the collaboration of many organizations working together to provide the best possible interfaces to developers.

Q: When is it inappropriate to reuse code?

Integrating IP, whether developed internally or externally, should never be considered a silver bullet. Code should be evaluated in terms of being the best solution for a problem and should not be reused in unintended ways or for the sake of reusing IP. IP should be assessed on its existing capabilities and not on the ability to modify its usage, which may result in productivity losses.

Additionally, the total cost of IP integration needs to be taken into account when deciding to incorporate. Project managers need to consider development costs, time-to-market benefits and maintenance costs to determine whether an IP reuse makes fiscal and technical sense for a given project.

Q: What cost models are useful in evaluating IP purchases?

When deciding to purchase IP, a team is presented with a classic build-versus-buy scenario. They need to consider whether they have the in-house expertise to develop the desired IP. Once the team has determined that they have or can acquire the necessary expertise, a proposal needs to be made on the time and internal cost of developing the IP, including validation and maintenance of the code base. Lastly, the impact on the overall project ship date must be factored in as an opportunity cost.

The cost of purchasing IP varies depending on the licensing model. IP vendors can license their IP as an up-front cost, royalty-based dependent on units or a combination of both. Additionally, service and maintenance charges may be part of an IP licensing agreement, which may offset some costs associated with bug fixes and IP integration. The design team must factor in these potential costs and weigh them against the cost of developing the IP internally as well as the potential time-to-market savings and reduced maintenance costs associated with purchasing IP. Oftentimes the issue boils down to the expertise of the team, the complexity of the IP and market pressures.

NATIONAL INSTRUMENTS™