

Design Tip #96 Think Like A Software Development Manager

By Warren Thornthwaite

For most organizations, the vast majority of users get to the DW/BI system through the BI applications, including standard reports, analytic applications, dashboards, and operational BI; all these applications provide a more structured, parameter driven, relatively simple means for people to find the information they need. In Design Tip #91 on marketing the DW/BI system, we described the BI applications as the product of the DW/BI system and emphasized that these products must be valuable, usable, functional, high quality, and they must perform well.

Most of these characteristics are formed in the BI application design and development process. Throughout application development, testing, documenting, and rollout, it is very helpful to pretend to be a professional development manager from a consumer software product company. Actually, it's not pretending. Real software development managers go through the same steps as the folks responsible for delivering the BI applications. The best software development managers have learned the same lessons:

- The project is 25 percent done when your developer gives you the first demo of the working application. The first demo from a proud developer is an important milestone that you should look forward to, but seasoned software development managers know that the developer has only passed the first unit test. The second 25 percent is making the application pass the complete system test, where all the units are working. The third 25 percent is validating and debugging the completed system in a simulated production environment. The final 25 percent is documenting and delivering the system into production.
- Don't believe developers who say their code is so beautiful that it is self-documenting. Every developer must stay on the project long enough to deliver complete, readable, high-quality documentation. This is especially true for any interactions or algorithms that are directly exposed to the users.
- Use a bug tracking system. Set up a branch in your bug tracking or problem reporting system to capture every system crash, every incorrect result, and every suggestion. A manager should scan this system every day, assigning priorities to the various problems. An application cannot be released to the user community if there are any open priority 1 bug reports.
- Place a very high professional premium on testing and bug reporting. Establish bug-finding awards. Have senior management praise these efforts. Make sure that the application developers are patient with business users and testers.
- Be proactive with the bug reports you collect from users and testers. Acknowledge receipt of every reported bug, allow the users and testers to see what priority you have assigned to their reports and what the resolution status of their reports is, and then fix all the bugs.

These lessons are especially important in organizations with large user communities. The same is true when creating operational BI applications that will be used by a large number of operational users. You will probably not meet everyone individually in these user communities, so your product had better be great.