Manufacturing Tech

Nick Konzel

Nick Konzel is the owner of Distributed Network Software. Retired from the U.S. Army as a logistician and earning an MIS degree from Penn State, The Behrend College, he has 20 years of executive management experience in manufacturing, distribution and service industries.

Complicated Software Development

Managing Software Projects
Change is the one constant in business today. Corporations find themselves continually prioritizing software investments with changing business demands. The technical nature of software development often times makes it difficult for decision makers to understand the software life cycle. Because analogies are helpful in explaining complicated topics, I will explain Custom Software Development by choosing an example we are familiar with.

Creating corporate software is a lot like building construction. When was the last time you witnessed new construction by a single person? Construction requires blueprints, tools, teamwork, and experience. And much like building software, finished construction requires maintenance and upkeep. What about changing critical business software as your business demands change? Think about the forethought and experience needed, in addition to the cost of changing plumbing in a building from half- inch to three-quarter inch piping one year after construction. Please allow me to explain.

Software development companies specialize in creating and maintaining custom software much like a contractor specializes in building and maintaining structures. The build team creates software to business specifications then reviews and adjusts the software to fit the business environment. The build team uses tools and processes to manage software creation. See some of the items listed below to explain a few critical software development processes and tools.

- Software Development Life Cycle (SDLC) – Yes, software has a life. It begins with a business idea, develops into meaningful processes and ends up as installed software that helps your business become more productive. The software can be as simple as a website or as complicated as an order-entry screen connected to an existing accounting system. In all cases the software begins new then ages as supporting technology becomes obsolete. Once this occurs the software needs a new ROI calculation and maintenance, thus beginning the extended life cycle. A good build team understands where the software stands in the SDLC and builds it with business patterns in mind to facilitate change.

- Software Project Management – It's always nice to know how long a project will take to complete and where the build team is in the SDLC. Build teams need tools to plan their time and construct software as business demands change. There are several tools that successful build teams use daily;

1. Project Tracker – this is a tool that allows the build team to help create design specifications (like building blueprints for a building). It estimates time (a.k.a. cost) to complete functionality and assigns team members to development tasks. It also provides business stakeholders and IT managers the much needed insight into project status.

2. Microsoft Visual Studio .NET (VS.NET) – this is an integrated development environment (IDE) or, in other words, software that allows developers to build new software. It's a tool box for the build team to better manage software delivery and maintenance by providing an integrated software platform and real-time reporting of key performance and quality metrics. The VS.NET IDE consists of several programming languages along with software deployment and database connectivity tools. It helps software build teams continuously improve their development processes. VS.NET IDE is widely accepted around the world as the premier set of software creation tools.

3. Cruise Control Build Software – this is untiring software that continuously compiles source-code into machine readable software [Source-code contains instructions that you and I can read; machine-readable code contains the same instructions that only a computer understands]. Cruise Control constant build software is a lot like the electrical fuse box in your building, constantly providing electricity which is attached to a fuse so that there are no electrical accidents. Cruise Control compiles and tests software before the build team releases it to the customer. This ensures that nothing breaks when changes are made to the software.

4. Tortoise SVN Version Control – this is a software repository that stores different versions of

production software as business needs are discovered. A critical function of any build team is keeping track of software versions. Tortoise SVN version control software allows build teams to store and roll-back all previous changes during the software creation process. It’s compatible with different IDEs and keeps historic changes to the build team work product much like an architect tracks construction changes using blueprint design software.

- Code review meetings – this is a peer review process for programmers. The source code is examined for efficiencies and design specifications. This pre-deployment meeting is for programmers to review the work of their peers. It’s a lot like a carpenter making sure that all studs in the wall frames are 16”-on-center before the drywall goes up (otherwise it would be a costly mistake to find out after the occupants moved in). A peer review looks at the internal software source code (the studs), not the end-user graphical part of software (curb appeal from the outside).

Deciding Whether to Build or Buy Corporate Software
Business leaders often ask, ‘should I buy or build my software?’ My response to them is to buy if possible. The responsibility of upgrades belongs to the owner (Microsoft, Google, and others all have SDLC plans and tools). Unfortunately, many off-the-shelf business software packages meet less than 80% of business’ needs and rarely connect to other enterprise software. On the other hand, if a business decides to build software they must be aware of the SDLC, tools, and build team processes for a successful custom software implementation. In either case it’s best to stick to your core business competencies and let the construction up to the contractor.

Having one or two employee software developers build and maintain corporate software without code review sessions or adhering to SDLC guidelines may be inexpensive in the beginning but can quickly overwhelm developers and make it nearly impossible to change corporate software without breaking important features that affect your business’ productivity. When contracting with a team of custom software developers, organizations are hiring a team of professionals, a team that uses SDLC tools and procedures, and a team that has the luxury of different team member’s skills. Team oriented software development ensures that business software is build to specifications, on time and produces the expected results for your company.