

1 Things You Might Not Know About

Agile Development

by Jeffery Payne

- 1** **AGILITY IS THE GOAL** The debate about what development practices are or are not agile is moot. Business stakeholders don't care what your development process is called; they just want a return on their software investment. Focus on becoming *more agile* instead of becoming "Agile" and your business value will improve.
- 2** **COLLOCATION OF TEAMS IS NOT NECESSARY** When building software, team rooms are definitely helpful for increasing agility but are not a showstopper. Distributed teams that focus on consistent, clear communication can use agile development practices to achieve business value.
- 3** **TEAMS ARE STILL ACCOUNTABLE** Implementing agile practices does not mean your software development teams are off the hook for on-time delivery. Agile greatly increases the visibility of day-to-day activities and drives teams toward higher productivity *and* accountability if implemented properly.
- 4** **BUSINESS SPONSORS NEEDN'T BE OVERBURDENED** Integrating business and domain knowledge into day-to-day project activities does not mean the process has to overburden your business sponsors. Business analysts can shoulder most of the day-to-day team interaction so business sponsors are called in only when really needed.
- 5** **UNIT TESTING IS NON-NEGOTIABLE** If you are not doing unit testing, you are not doing agile development. Unit testing is a cornerstone practice that makes other practices—such as continuous integration and refactoring—possible and valuable.
- 6** **AGILE SOFTWARE CAN BE SECURE** There is nothing about agile development that results in insecure software. Agile done right integrates secure development practices and security assurance into all development and testing activities.
- 7** **AGILE IS RIGOROUS** Don't be fooled into thinking that agile is a less-rigorous process than other development approaches. When done correctly, agile is a consistent, repeatable process with strong measurement of progress, risks, and productivity.
- 8** **GURUS DON'T HAVE ALL THE ANSWERS** Software succeeds or fails in the trenches where it is built. Many agile coaches haven't written a commercial software application in a decade. Trust and learn from those who *do* agile, not just talk about it.
- 9** **ARCHITECTURE AND DESIGN ARE STILL NECESSARY** You can't refactor your way into an initial architecture. Time must be spent during initial planning and iterations to get your architecture solid. From there, incremental development that includes refactoring can drive your software to completion.
- 10** **DOING DOCUMENTATION IS OK** Many software products must comply with standards, regulations, and corporate development policies. There is no reason that agile cannot be used to build these types of products as long as the appropriate documentation is created during the process.