Brady Field

A Review of Formal Methods

**Important points**

“The underlying philosophy of formal methods has not changed drastically in a quarter century. Nevertheless, this approach is a revolutionary paradigm shift from conventional notions about computer programming. Many software engineers may have adopted the new methodologies without fully understanding or even being aware of the root concepts.” [323]

“Formal methods are an adoption of the axiomatic method, as developed by these trends in mathematics, for software engineering (MacKenzie 2001). In fact, Edsger Dijkstra (1989) suggested, somewhat tongue-in-cheek, that computer science be renamed Very Large Scale Application of Logic (VLSAL).” [324]

“Formal methods can be used to specify aspects of a system other than functionality. For example formal methods are sometimes applied in practice to ensure software safety and security properties of computer programs.” [325]

“Formal methods suffer from certain limitations. Some of these limitations are inherent and will never be overcome. Other restrictions, with research and practice, will be removed as formal methods are transitioned into wider use.” [326]

“Successful projects were often successful because of the role of one or two key exceptional designers. These designers had a deep understanding of the applications domain and could map the applications requirements to computer science concerns. These findings suggest the reduction of informal application knowledge to a rigorous specification is a key problem area in the development of large systems.” [327]

“Although no prominent advocate of formal methods recommends that testing be avoided entirely, it is unclear what role testing can play in increasing confidence in the areas not addressed by formal methods. The areas addressed by testing and formal methods may overlap, depending on the specific methodologies employed.” [328]

“No matter to what extent an organization decides to adopt formal methods, if at all, training and education issues arise. Many programmers have either not been exposed to the needed mathematical background or do not use it in their day-to-day practice. Even those who thoroughly understand the mathematics may have never realized its applicability to software development.” [329]

“In general, formal methods provide for more precise specifications. Misunderstandings and bugs can be discovered earlier in the life cycle. Since the earlier a fault is detected, the cheaper it can be removed, formal specification methods can dramatically improve both productivity and quality. How best to use formal methods in a specific environment will only be determined through experimentation.” [330]

“An automated verification system provides a means for the user to demonstrate the existence of a formal proof of a software system.” [331]

“Even if formal methods are not integrated into an organization’s process, they can still have positive benefits.” [332]

“The full-scale use, transition, and cost-effective use of formal methods is not fully understood. An organization whose leaders can figure out how to effectivily integrate formal methods into their software processes will be likely to produce higher-quality software and thereby gain a competitive advantage.” [333]

**Disagreements**

“The benefits of proving that unsafe states cannot arise, or that security is assured, can justify the cost of complete formal verifications of the relevant portions of a software system.” [325]

The author asserts that in using these formal methods from a security perspective will result in a product where unsafe states do not occur and security in the product is assured. This is an impossible claim. There is no such thing as a perfectly secure piece of software. Given enough resources an attacker can always compromise the system. The job of securing systems is to make the cost of compromising the system more trouble than it is worth for the attackers. This is a faulty claim by the author.

“These programmers will know that programs can be developed to be fault free from the first execution.” [332]

Fault free software doesn’t exist. This is an inherent characteristic of software.

**Questions**

I understood everything in the article.