

The Role of Formalism in System Requirements

RE'23 Journal First Submission

Authors

[Jean-Michel Bruel](#)

[Sophie Ebersold](#)

[Florian Galinier](#)

[Manuel Mazzara](#)

[Alexandr Naumchev](#)

[Bertrand Meyer](#)

Abstract

A major determinant of the quality of software systems is the quality of their requirements, which should be both understandable and precise. Most requirements are written in natural language, which is good for understandability but lacks precision.

To make requirements precise, researchers have for years advocated the use of mathematics-based notations and methods, known as “formal.” Many exist, differing in their style, scope, and applicability. The present survey discusses some of the main formal approaches and compares them to informal methods.

The analysis uses a set of nine complementary criteria, such as level of abstraction, tool availability, and traceability support. It classifies the approaches into five categories based on their principal style for specifying requirements: natural-language, semi-formal, automata/graphs, mathematical, and seamless (programming-language-based). It includes examples from all of these categories, altogether 21 different approaches, including for example SysML, Relax, Eiffel, Event-B, and Alloy.

The review discusses a number of open questions, including seamlessness, the role of tools and education, and how to make industrial applications benefit more from the contributions of formal approaches.

Published in June 2022 in [ACM Computing Surveys 54\(5\)](#): 93:1-93:36.

Link to the **online version** at ACM: <https://dl.acm.org/doi/10.1145/3448975>