1. Important Points
   1. “Design is both ‘the process of defining the architecture, components, interfaces, and other characteristics of a system or component’ and ‘the result of that process.’” [195]
   2. Software design produces “a description of the software’s internal structure that will serve as the basis for its construction.” [195]
   3. “A specific solution can be considered *good or* *bad,* not *true or false*.” [195]
   4. “Design in its general sense, can be understood in terms of five key concepts: goals, constraints, alternatives, representations, and solutions.” [195]
   5. Software requirements analysis, software coding and testing, and software integration and qualification are “software development activities that are more directly coupled with software design.” [196]
   6. Software architecture is “the fundamental organization of a system embodied in its components, their relationships to each other, and to the environment, and the principles guiding its design and evolution.” [196]
   7. Views are “high-level facets, or perspectives, of a software design.” [196] “A view ‘represents a partial aspect of a software architecture that shows specific properties of a software system.’ It is ‘a collection of models that represent on aspect of an entire system.’”[197]
   8. “The overall organization of an SDD [should] be obtained as the composition of a number of ‘design views.’” [197]
   9. “They key goal of patterns is … to codify and document … those commonly recurring solutions to typical problems.” [198]
   10. “An architectural style can be seen as a meta-model that provides a software system’s high-level organization—its *macro*architecture. [sic]” [198]
   11. “Design patterns are used to describe details at a lower and more local level—the *micro*architecture. [sic]” [198]
   12. “An important goal of software design has always been to allow for the *reuse* of software elements.” [199]
   13. “Software quality … can be characterized by the following six properties: functionality, reliability, usability, efficiency, maintainability, and portability.” [199]
   14. Measures, such as “function-oriented (structured) measures” [200] and “Object-oriented measures,” [200] “can be defined to obtain *quantitative estimates* of a design’s size, structure, or quality.”
   15. “Software design reviews” and “simulation and prototyping” are “other techniques [that may] be used to evaluate the quality of a design.” [200]
   16. “UML (Unified Modeling Language) has become an almost de facto standard for software development notations.” [201]
   17. Structural descriptions give a static view of the software. [201] Behavioral descriptions “describe the dynamic behavior of systems and components.” [201]
   18. “A key characteristic of any interface specification … is that it should be a *two-way* description: what the element *provides* and what it *requires*.” [202]
   19. “General strategies and enabling techniques” include abstraction; coupling and cohesion; divide and conquer; information hiding and encapsulation; and sufficiency, completeness, and primitiveness. [202]
   20. “*Structured (function-oriented) design* centers on identifying the major systems [sic] *functions,* which are then elaborated and refined in a top-down manner.” [203]
   21. “Key concepts of structured design are those of coupling and cohesion.” [203]
   22. “An object can be created/destroyed, has a unique (immutable) *identity*, possesses a (mutable) *state* (i.e., it evolves in time), and exhibits some well-defined *behavior* through services it offers.” [204]
   23. “OO design methods aim at developing software systems composed of interacting objects that are highly modular and … [are] easy to modify, extend, and maintain.” [204]
   24. “How objects from the various classes collaborate to provide the desired system behavior is described using *interaction diagrams*.” [205]
   25. “*Data-structure-oriented design* … is an approach in which the emphasis is on the data that a program manipulates rather than the functions it performs.” [206]
2. Disagreements

“A number of measures can be defined to obtain quantitative estimates of a design’s size, structure, or quality.” [200]

I disagree with the authors’ assertion that quantitative measures can estimate design quality. While structure charts or class diagrams provide good representations of size and structure, these diagrams cannot meaningfully inform on design quality. Attempting to quantify quality by observing its structure or class diagrams will only provide a shallow glance at quality. My disagreement does not hold when working with a design structure that is obviously bad.

1. Questions and Clarifications
   1. I have no questions about this article.