1.6.2

Though building architectures can be considered an abstraction of a building's physical properties in the same as software can be considered an abstracted way of manipulating logic atop physically-implementing logic-inducing circuitry, the general context associated with software is vastly different from that of buildings. Namely, materials, uses, processes, properties, and commonly familiar conceptions approximating the appropriateness of types of design for each specific case may not be as readily accessible for the public/architects working with software as with buildings.

1.7.1

Hand-drawn sketches and blueprints of buildings may resemble, possibly, activity-, use-case, or class-type diagrams used for software development to aid during conceptualization; while in a more refined way the implementation described in such may be further detailed using prose and mathematical formulaic symbolisms to aid in conceiving specific algorithms likely to be employed.

1.7.6

Software architecture, in that it is an abstract representation of the notion of a desired piece of software or an expression of such via the various artifacts and conceptions produced in association with a software implementation's development, is generally considered to be distinct from any notion of physicality of an engineered product. Rather, like other, more intellectually sophisticated fields, such a law, medicine, automotive engineering, or highly-reputed envisionary-workers in the realm of the arts, software architecting requires more reliance on vague notions of process and craft to include repurposing or reconstructing intentions (often gleaned by applying contextual insight held by the architectural artisan, himself) from stakeholders into products that are functionally satisfactory according to the expressed given thereby. Much the same as a doctor practicing medicine or a visionary crafting a piece of art may tailor the process and techniques to suit the particular needs of the stakeholder concerned, a software architect will often apply logic that is specific to the domain or circumstance with which he or she is dealing. As concerns superiority of a given field of practice's analogy to software architecture, as compared with the analogy of building architecture to it, none may necessarily outperform that, but each may have its own advantages. For instance, as compared to Michaelangelo's commissioned-works case, the software architect may parallel his processes more so than a building architect's in that there are unprecedented and endless transformations of color, proportionality, and realization to satisfy the artistic desires of the customer that a building architect would not be advised to attempt to parallel in his craft, for the engineers following his blueprint would be angry or baffled, at the least to follow blueprints violating understood principles of physics. In automotive engineering, or essentially any engineering discipline in fact, principles constituting the primary support structure of the field may deem it a violation of software engineering's practice, whereas proponents of software engineering likening the field more to science than engineering will attest. Further, just-in-time concepts may not withstand the pressures of the SOWA development lifecycle. As with law, processes are created and later interpreted and enforced by humans with in turn inviolates particular notions of software architecture that lay the foundation for building further-

cognitively complex advancements in the field, such as that machines interpret programmer's code, subject to the decidability quagmire; however, with software while there are few programs that are tractably proveable, at least some programs by logic are. As law is a discipline rooted in ethics, is it the case that there must not be anything that is proveable barring initially agreeing upon some fundamental baseline of human decency or correctness, whereas with a building's design we can measure easily the building's ability to stand the test of time and natural phenomenon, precisely, or a software product's ability to produce a correct answer given specific inputs. Medical practictioners have been likened to software architects in the application of processes that may fall under the 'Zen', 'Tao'–type genres or a predominantly biological- [and/]or chemical- [hybrid-]based approach , these analogies do not outperform those of building architects to software architects, as medical practitioning subject to the rules of these techniques may require assumptions in process or application that deem it more meta-physical or strictly physical as compared with software or building architecture craftsmanship.