Two Kinds of Engineering Paradigms (See FIG-1&2 Below)

The engineering paradigms for designing and building large or complex products can be broadly categorised in to two kinds. First one is ***non-component-based*** engineering paradigm, where each product is designed by using various kinds of **reusable** ingredient parts such as cement, steel, plastic, wood, leather, chemicals, bricks, nickel, acid for lead-acid car battery, paints, silicon & metal-oxides for IC-chips, metals, and alloys to name a few. For example, design and construction of products such as houses and buildings is an example for non-component-based engineering paradigm, where it is not possible to disassemble the product. Both Civil and exiting software engineering paradigms are examples for non-component-based paradigm.

Second is ***component-based*** engineering paradigm, where each product is designed by building and assembling multiple components, and the product can be disassembled. In case of *component-based paradigm*, each elementary component is designed and built by using various kinds of **reusable** ingredient parts such as steel, plastic, nickel, acid for lead-acid car batteries, paints, silicon and metal-oxides for IC-chips, glass, cloth, metals and alloys. Multiple elementary components may be used to build container components. Both paradigms may use same reusable ingredient parts, where second paradigm builds components by using reusable ingredient parts.

Engineering paradigms such as mechanical, aerospace, computer and electronic engineering disciplines are examples for component-based paradigm, since each of the large products such as cars, computers, airplanes, jets, machines and machinery for factories are built by assembling components, and the product can be disassembled. Each of the elementary components for all these products is built by using reusable ingredient parts. Hence, the paradigms for both kinds try to maximise the reuse, but component-based paradigm also try to maximize modularization.

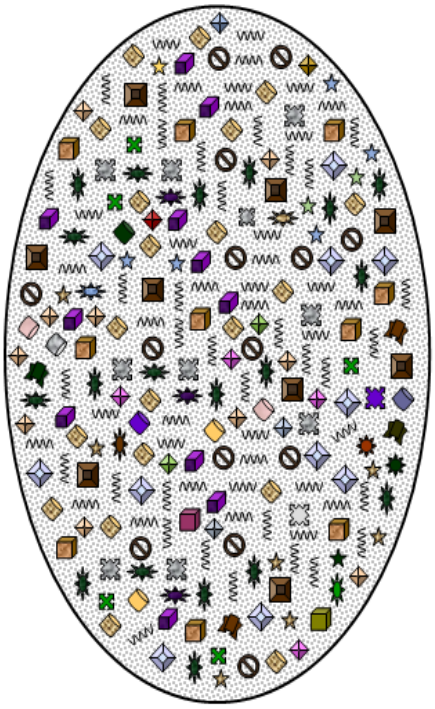
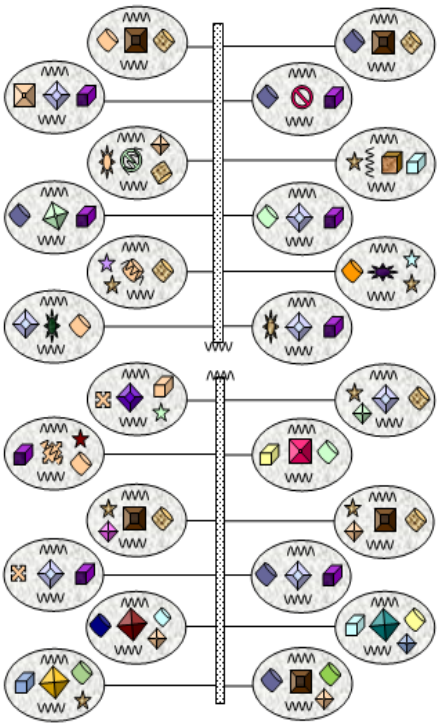
FIG-1: Total Lines of Custom Application Code 150K (After using every possible reusable library/API)	**FIG-2**: Total Lines of Code 50*3000 (=150K) About 50 modules of average size of 3K lines
	

FIG-1: Illustrates a product built by using every possible kind of reusable parts (shown in various shapes), except using specific kind of parts that can be assembled and disassembled, where such specific kind of parts are widely known as components (in the context of countless product we know and use). In other words, this product is not built by assembling such components.

FIG-2: Illustrates a product that is built by plugging-in numerous components (i.e. components are very specific kind of parts that can be assembled and disassembled), where each of the pluggable (and easily unpluggable) components is designed and built by using every possible kind of reusable parts (shown in various shapes). Both products use same number, amount and/or kind of reusable parts.

Only difference is that: (i) the second is partitioned in to self-contained modules of components, (ii) each of the components is designed and built (and tested) individually, and (iii) the product can be created in minutes by plugging-in all the components in to a SoA, once all components are ready & tested individually.

Patents granted for real-CBD by USPTO so far: 7827527, 7840937, 8527943, 8392877, 8578329 and 9058177 (not including pending patents). Searching in Google for ***Raju Chiluvuri patents*** returns pages: https://patents.justia.com/inventor/raju-v-chiluvuri & https://patents.justia.com/inventor/raju-chiluvuri