

Discussion Board 3

Aidan Polivka

5/16/2024

I would argue that maintainability is the most important “-ibility” in systems engineering. Of the 6 “-ibilities” outlined on the discussion board prompt (Feasibility, Reliability, Usability, Maintainability, Producibility, Disposability), strong maintainability of a system best mitigates the pitfalls of a system that is weak in the other “ibilities”.

Let’s start by talking about what a “maintainable” system is. I think a maintainable system is well built, well documented, and open for expansion. A maintainable system is loosely coupled to third party software, making them simpler to replace. A maintainable system’s sub-systems are also loosely coupled between one another, allowing for modularization and sub-system refactoring. Using the standard project example, the employee time management sub-system should be self-contained and able to be easily refactored without breaking the inventory or point of sale systems.

I think one of the strongest challengers to maintainability’s importance is Feasibility. With poor feasibility, a system possibly can’t even be built. Poor feasibility means that the requirements established by the business aren’t easy or cost effective to implement. However, with a well-built, maintainable system, difficult problems or system expansions are much more manageable. Even if the problem isn’t feasible, a well-built system has the best shot at supporting something close to what the client wants.

With a maintainable system, if the reliability of the system is suffering it’ll be easier to fix the unreliable parts of the system. A maintainable system is well documented and built to scale and expand. So, in a buggy system it should be easy to identify and resolve bugs, or even refactor and rebuild parts of the system. Additionally, a maintainable system’s components are self-contained and loosely coupled. This means that the unreliable parts of a system can be fixed without impacting seemingly unrelated parts of the system.

If the system lacks usability but is strong in maintainability, the parts that aren’t user friendly are well documented and are more easily refactored. If the system was well documented and built in a way that is option enabling, it should be easy to identify and implement refactoring opportunities. This problem again benefits from the loose coupling of a well built system in the same way that an unreliable system does.

Producibility in my mind is the second most important “ibility”. If procedure isn’t well defined, and the development process isn’t suitable to keep the project moving at a strong pace then stakeholder trust is going to suffer. Without good faith from the stakeholder and good project velocity, the project risks being abandoned. It won’t matter how maintainable the system is if the project is shut down. However, building a system that can pivot, is well built, and well documented will more easily allow for a project development restructuring.

The final “ibility” is disposability. This concept refers to how easily a system can be disassembled. I would think that a strongly disposable system would be well modularized, and components would be easily swappable. I think this concept is also true for a maintainable system. A maintainable system should also be disposable in that its components can be replaced with new components.