Topic: Dependency Injection

In software engineering, dependency injection is a technique whereby one object (or static method) supplies the dependencies of another object (Karia, 2018). Dependency Injection refers to the ability to pass dependencies when they are required instead of initializing the dependencies inside of the recipient class Dependency injection takes over the responsibility of creating objects, knowing which classes require the objects, and providing the objects. This allows you to change objects at Runtime, rather than compile time. Dependency Injection uses the concept of “Inversion of Control”, which states that classes should not configure dependencies statically, but instead using some other outside entity. Some. Dependency Injection can help with unit testing, reducing boiler plate code, extending applications, and enabling loose coupling (Karia, 2018). Coupling refers to the amount of direct knowledge objects have of each other. Loose coupling means the objects are largely independent. The disadvantages are it can push compile errors to runtime errors, which are often harder to debug.

Karia, B. (2018, October 18). A quick intro to dependency injection: What it is, and when to use it. freeCodeCamp.org. <https://www.freecodecamp.org/news/a-quick-intro-to-dependency-> injection-what-it-is-and-when-to-use-it-7578c84fa88f/