475 Software Engineering for Industry: Topic 1 : Legacy Code

Part I:

Legacy code refers to the code written by predecessors in a broad sense. Compared with non-legacy code, the code is usually lacking some corresponding tests[1]. Most software developers are reluctant to process and modify strange legacy code. However, many companies still have a number of legacy codes. The following will explain why so many companies have legacy code, although no-one likes to use it.

- Everything is evolving and changing, including software. The Software will automatically switch to maintenance mode if the version control stops at some point. Once the software enters the maintenance phase, it becomes legacy [4].
- In the development of some projects, it is difficult to discard legacy code directly, especially to develop systems with quite complicated business logic, such as some financial telecommunications systems. Obviously, the legacy code can complete the system function throughout the business time. If we abandon it directly, redevelopment will cost a lot of resources and is not necessarily successful. If the new requirements change, there is often no time to redevelop the code.
- Experiencing with legacy code is unavoided for every developer. Everyone is capable of working in a fresh environment where everything is new while using legacy code is quite a difficult task. However, writing legacy code can hit some important deadline due to a range of business reasons [5].
- From the perspective of the company's cost, using legacy code can sometimes yield better returns. Money spent on a quick-and-dirty project that allows immediate entry into the market may be better spent than money spent on designing better structure [2]. As long as the initial release is below the design pay-off line introduced in the design Stamina Hypothesis. Because sometimes it is better to trade off design quality for speed [6].
- In addition, there is often a phenomenon of employee turnover in the workplace. Once the person responsible for the software being developed leaves, the rest of the code naturally becomes legacy code.

Reference:

1. Michael Feathers, Working Effectively with Legacy Code, Upper Saddle River, N.J.: Prentice Hall PTR 2004
2. Foote, Yoder, Big Ball of Mud, University of Illinois at Urbana-Champaign 1304 W. Springfield Urbana, IL 61801 USA
3. https://simpleprogrammer.com/deal-with-legacy-code/
4. https://blog.rinatussenov.com/what-is-legacy-code-is-it-good-for-you-fb260a467fb7
5. https://martinfowler.com/bliki/TechnicalDebt.html
6. https://martinfowler.com/bliki/DesignStaminaHypothesis.html