

# Why change is needed in software

[Technology](#), [Information Technology](#)



WHY CHANGE IS NEEDED IN SOFTWARE By Introduction A software system that is utilized in a real-world setting has to change or become increasingly less useful due to technological advancement. Software change is very significant since organizations are completely reliant on their software systems and have spent millions of funds in the systems (Reifer 2011). Real world setting is involves incessant change thus software applicable must continuously change to keep up with demand and accuracy. Corporations' systems are important business assets and companies must invest in system change to sustain the worth of these assets (Neufelder 1992).

#### Reasons for change

There are several reasons for software change. New advances in innovation may render an existing piece of system obsolete thus change is inevitable. For instance, software needs to be capable to adapt to fresh hardware or peripherals (Schlesinger 2010). In case there is no communication between software and hardware change is always the advisable solution. It is no good purchasing a new scanner or printer if existing software cannot interface with it. Technology advancement makes each and every day novel systems come to pass (Reifer 2011). They are a lot of other software, which are not in the system currently and individuals would ask why such software is no longer applicable. This is due to inability to tally with the new situation and hence necessitates to be wiped away. It should always change, with illustration it will render non-functional and cannot serve any better reasons rather than the system (Neufelder 1992).

Software maintenance is also another reason that is associated with system

change. Changes to the software are executed in reaction to changed necessities but the basic structure of the software remains constant. This is the most general strategy applied in system change. System change is essential in maintenance of its function and reliability (Naveda and Seidman 2006).

Architectural transformation is associated with technological advancements that drive change initiatives to ensure the functionality of software is adjusted to meet real-time changes. This is a more radical strategy to system change than sustenance as it involves making important changes to the structural plan of the software system (Burge 2008). Most frequently, systems transform from a centralized, data-centric structural design to client-server design. In such scenario, change is necessary to meet the architectural transformations (Hallsteinsen and Paci 1997).

Software change is also associated with software re-engineering. Software re-engineering is different from other approaches in that no innovative functionality is extra to the system. Rather, the classification is customized to make it easier to appreciate and revolutionize. System re-engineering may engage some structural modifications but does not normally architectural change (Schlesinger 2010).

## Conclusion

System change is essential in ensuring that real time requirements are achieved as necessary. New advances in innovation may render an existing piece of system obsolete thus change is inevitable. For instance, software needs to be capable to adapt to fresh hardware or peripherals. Changes to the software are executed in reaction to changed necessities but the basic

structure of the software remains constant.

#### Reference List

- BURGE, J. E. (2008). Rationale-based software engineering. Berlin, Springer.  
<http://public.eblib.com/EBLPublic/PublicView.do?ptiID=364090>.
- HALLSTEINSEN, S., & PACI, M. (1997). Experiences in software evolution and reuse: twelve real world projects. Berlin [u. a.], Springer.
- NAVEDA, J. F., & SEIDMAN, S. B. (2006). IEEE Computer Society real world software engineering problems a self-study guide for todays software professional. Hoboken, NJ, John Wiley & Sons. <http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=158220>.
- NEUFELDER, M. A. (1992). Ensuring Software Reliability: Quality and Reliability. CRC Press
- REIFER, J. D. (2011). Software Change Management: Case Studies and Practical Advice: Case Studies and Practical Advice. Microsoft Press
- SCHLESINGER, R. (2010). Developing real world software. Sudbury, Mass, Jones and Bartlett Publishers.