LITERATURE SURVEY

TITLE OF THE PAPER	AUTHOR	OBJECTIVES	METHODOLOGY	FINDINGS	REMARKS
An Approach for Component-based Software Development	Fu Lingyun, Sun Guang,et al.	A semi-formal method for component-based software development with the assumption that components are viewed as black box. It restrict concentration on the coupling degree between components and the cohesion of the components themselves.	For the sake of controlling the coupling degree between components, make sure the components at the same level have no interaction with each other. This process is similar with creating a graph or a tree. So, use interaction graph or isomorphic tree of the graph to express the component composition.	For improving the cohesion of component and weakening the coupling between components, we transform the graph into a tree. It can simplify the System development and provide a system research method with graph and tree theories.	• It restrict the concentration on the coupling degree between components and the cohesion of the components themselves. • Reliability issues.
Seamless Code Reuse with Source Code Corpus	Tetsuo Yamamoto Norihiro Yoshida, et al.	A technique to support code reuse on demand without suspending coding tasks. The key to search code is half-written code. Users do not have to consider keywords for searching code. Consequently, the time required for code reuse is much shorter than keyword-based code search systems.	Two processes- One is corpus creation process, which creates a code corpus from target source files. The created corpus is stored into a database. The other is corpus usage process, which suggests reusable code based on developers reuse requests. When a programmer is implementing a functionality, code stored in the corpus is recommended on an integrated development environment that the programmer are using. The two processes are independent, both the processes function separately.	When searching a reusable code all the codes will be available but sometimes its not suitable for our requirements .so, a technique is needed to makes the code more suitable for project.	For effective transformation rule, partition points between key sequences and value sequences, and key sequences matching rules in the corpus. The future plan is to develop more suitable ranking methods for providing appropriate reusable code.
Building a Community System to Teach Collaborative Software Development	Modi Lakulu, Rusli Abdullah, et al.	• To gain experience by working on real world projects for students. • To make projects at low cost by providing open source projects for organization.	• It uses the framework KMSOS2oD consisting of prototype approach. • The main parts of KMSOS2oD framework is knowledge sharing and smooth communication part. • Knowledge sharing means experience of each developer will be shared among all other developers. • Smooth communicatio	• Complexity of real world projects. • Lacking of code reusability options.	• Organizational knowledge management system. • Using of 3 layer approach client, expert team, developing community. • Efficient use of database makes the code reusability more reliable.