Dependency analysis tools

A technique used to analyze the dependencies between activities

* it enables us to describe interdependence between activities
* it is based on information, material
* it is used to describe the order of the execution of activities

Procedures:

1. Identify the steps in process
2. Determine the relationship between the steps
3. Chart the dependencies in an activity dependency diagram to show predecessor and successor relationship between the steps

Available tools:

* Understand - https://scitools.com/feature/combined-language-analysis/

With Combined Language Analysis, ***Understand*** can handle code bases written in multiple languages, such as C++ and Java, or Ada combined with FORTRAN.

Most large projects involve more than one programming language. For instance, Ada calling C, which calls Java, and then the Ada calls FORTRAN for Math functions.

***Understand’s*** Combined Language Analysis feature follows jumps between languages so that you can get consolidated information about the complete system showing dependencies between parts of the code written in different languages.

For an example of this, see the ZLib project that we ship as an example project (Help -> Example Projects). It has C, C++, C#, Assembly, Ada, Delphi, and Pascal.

* **eDepend - https://marketplace.eclipse.org/content/edepend-graphical-dependency-analysis-tool/metrics#group-details**

Plugin for Eckuose, graphical and interactive in real-time dependency analysis.

It is made for Java, it integrates a rich set of tools to effectively detect, display, navigate and analyze class/package/project dependencies

For a project leader this tool helps to control and supervise the quality of codes. As quality engineer it gives you an overview over the architecture of an projects.

There are many others tools such as:

* [Ant](http://jakarta.apache.org/ant/index.html)
* [Enterprise Module Management Agent](http://sourceforge.net/projects/emma/)
* [Make](http://www.gnu.org/software/make/make.html)
* [RPM Wizard](http://sourceforge.net/projects/rpmwiz/)
* [package dependency viewer for FreeBSD](http://sourceforge.net/projects/gpkgdep/)
* [Package Tools](http://sourceforge.net/projects/pkgtools/)
* [DBDependency](http://sourceforge.net/projects/dbdependency/)
* [Dependency Finder](http://sourceforge.net/projects/depfind/)
* [Dependency Walker](http://www.dependencywalker.com/)
* [Flow and Dependency Analysis of SQL in SQL Scripts](http://wiki.cs.uiuc.edu/cs497rej/Flow+and+Dependency+Analysis+of+SQL+in+SQL+Scripts)
* [JDepend](http://www.clarkware.com/software/JDepend.html) - JDepend traverses Java class and source file directories and generates design quality metrics for each Java package. JDepend allows you to automatically measure the quality of a design in terms of its extensibility, reusability, and maintainability to effectively manage and control package dependencies.
* [jmove](http://sourceforge.net/projects/jmove/)
* [Makedep](http://sourceforge.net/projects/makedep/)
* [NetGraph](http://sourceforge.net/projects/netgraph/)
* [The Dependency Inversion Principle](http://www.objectmentor.com/publications/dip.pdf)

**Jdependency** - https://github.com/tcurdt/jdependency

jdependency is small library that helps you analyze class level dependencies, clashes and missing classes.

What it can do:

* Finding classpath clashes
* Finding missing classes
* Finding unused classes