CS 499/599 Research Methods for Software Engineering

Research Project

D1. Research Design

Group 1: vg588@nau.edu, kv582@nau.edu, au282@nau.edu, sv799@nau.edu

1. Research topic:

Developers depend on many packages in their project to make use of already developed functionality. Every project needs to be up to date with the latest dependencies for it to have the latest features and to avoid compatibility issues. Our research topic mainly focuses on the analysis of project dependencies and its internal dependencies. It addresses a problem commonly faced by many developers/managers which deals with the problems related to package dependency upgradation or deprecation.

2. Research Problem:

Developers rely on an increasingly high number of packages to build their programs, reusing of code increases productivity, improves software quality and decrease time-to-market. However, this kind of development creates a lot of dependencies on packages, sometimes these packages could be deprecated and managing these deprecated packages leads to serious issues which needs to be addressed.

Also, developers or organization encounter difficulties with dependencies, when amended to the latest version of packages. Upgrading from an older to newer version would require more resources, extra effort of development and QA which increases the cost for the organization. If the information regarding updated or outdated versions were not informed to the developers/managers, that could lead to the issue of malfunctioning of software in future or immediately.

3. Evidence:

An example incident in npm (Node Package Manager) is the release of a backward incompatible minor version 1.7.0 of the package "underscore" that caused many complaints among dependent packages about underscore not respecting Semantic Versioning. Another anecdote is the removal of the "left-pad" package that caused widespread breakage among big internet sites like Facebook, AirBnB, and Netflix. Also, a survey of 2,000 developers across different ecosystems has reported that 70% of the Node.JS developers have experienced breaking changes caused from updates when building their package.

References:

- Javan Jafari, D. E. Costa, R. Abdalkareem, E. Shihab and N. Tsantalis, "Dependency Smells in JavaScript Projects," in IEEE Transactions on Software Engineering, doi: 10.1109/TSE.2021.3106247.

- https://medium.com/coorva/a-simple-strategy-to-manage-your-javascript-projects-dependencies-d413f3d0ed2f

4. Purpose statement:

The main purpose of this analysis is to mitigate the problem faced by developers/naive programmers to resolve outdated/deprecated package dependency issues or incompatibility of latest dependency version with the existing dependencies.

5. Research questions:

RQ 1: What is the health of project dependency involved in the project?
RQ 2: What are the compatibility issues involved in the upgradation of project dependencies in the future?

6. Objective:

The objective of the project is to identify the dependency of deprecated packages. Programmers often won't be intimated regarding the deprecated packages in a project which often leads to a problem. The severity of the problem increases if the project is entirely depending on these packages. In this case, the code has to be completely modified or needs inclusion of other/similar functionality.

Developers will have an idea regarding the deprecation of packages as they will be frequently working with the project code. They can easily find out the information like current version of the package and also get suggestions from package manager regarding the latest version of the package to be installed if the developer wants to. This kind of information won't be available to the naive programmers as they won't have idea on the package manager and the project dependencies which creates an issue in the long run. The project managers won't be aware of the deprecated packages or latest packages available in the package manager. Our goal is to find the dependency risk involved on these packages.

7. Scope:

We will be restricting our scope to JavaScript/TypeScript projects for the package dependency analysis. For data, we will be considering the GitHub projects which contains more than 70% of JavaScript/Typescript code. Our research scope will resolve up to 3-4 levels of package dependencies or internal package dependencies.