Universidad de los Andes Colombia

Faculty of Engineering

Systems and Computing Engineering Department

CollabIDE

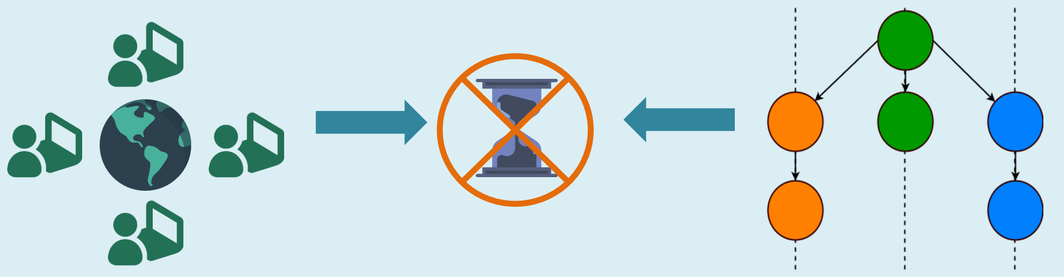
Santiago Beltrán Caicedo(s.beltran10@uniandes.edu.co), Nicolás Cardozo (n.cardozo@uniandes.edu.co)

PROBLEM

Decrease in productivity caused by *Overhead* generated from interruptions of the coding workflow. These interruptions come from activities related to version control.

Contributions from various developers in a ***distributed development environment.***

Product variants in ***software product lines.***



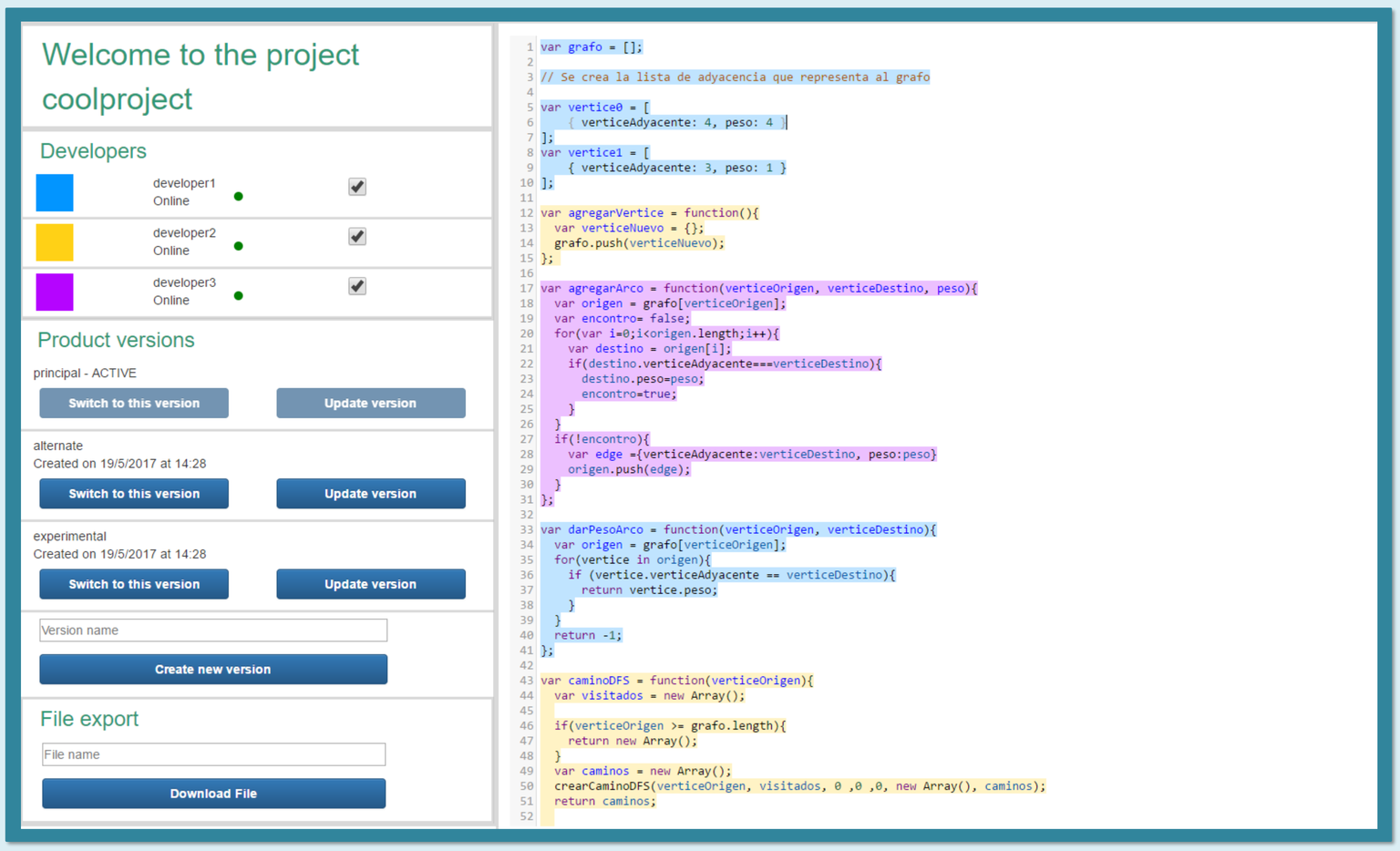
How do we solve it?

Build a collaborative IDE. The IDE must have features that reduce the *Overhead* generated by workflow interruptions. The requirements of the IDE: **contribution management**, **version management** and **concurrent development**

Context oriented programming provides elements that are necessary for implementing the requirements of the IDE, mainly because handling different contexts (developers and product versions) must be done.

COP

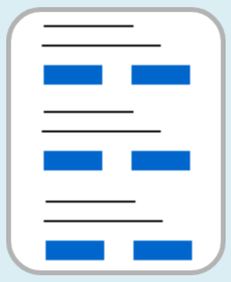
CollabIDE



Contribution management

Identifying authors of code fragments and the possibility of toggling which ones are visible help in conflict resolution.

Version management



The time required for creating and switching between product versions is reduced. **Every developer obtains a new product version** the moment it is created by any developer.

Concurrent development

Developers can **view in real time all the changes made in the IDE** (Product versions and code). This feature eliminates the need of constantly having to obtain the latest changes made by other team members.

Validation

2 Experiments

- eclipse + git VS **CollabIDE**

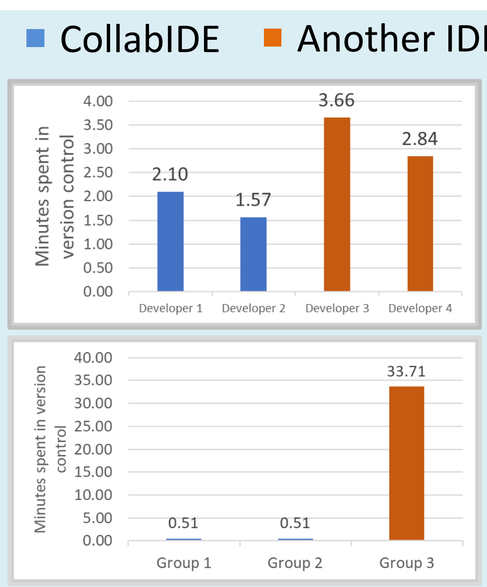
  4 developers

  Product variants development

- Sublime Text + git VS **CollabIDE**

  3 groups – 2 developers per group

  Collaborative product development



The graphs show the time spent in version control for each experiment. In both cases the *Overhead* was reduced when using CollabIDE.