

## History of Version Control

### What is Version Control?

Version Control exists in many forms, the most basic of which is practiced by almost every college student today, redundant copy storage. Whenever you save a rough draft, or “older version”, of a school paper you are practicing a very basic form of version control. A Version Control System is defined as the management and storage of files that allows for concurrent modification among programmers and historical documentation.

### What do modern Version Control Systems offer?

Version Control Systems play an important role in large scale software projects involving multiple developers.

Programmers rely on version control systems to accomplish the following tasks:

- To share files concurrently
- To go back in time to fix bugs
- Safely experiment with new ideas without altering known working solutions
- To share the latest and greatest versions by merging their local copies in a repository
- It can even be used to help understand how a project has evolved from its inception to its most modern form. It is in this area, however, that current forms of Version Control come up short.

## The Problem

Version control lacks the ability to allow a programmer to see how the project evolved. The time between committed versions is too long. Some developers commit once a week, once a day, or once an hour, these commits are only snapshots of what happened between the time of one commit and the next. There is no way to animate the evolution of files because too much information is lost between commits.

Still photos are similar to versions in a version control system. They can tell where we are but not how we got there.

Similar to a flip book, an individual image can't be animated but once flipped through with the rest, a story is created.



## Project Ali

### Objective

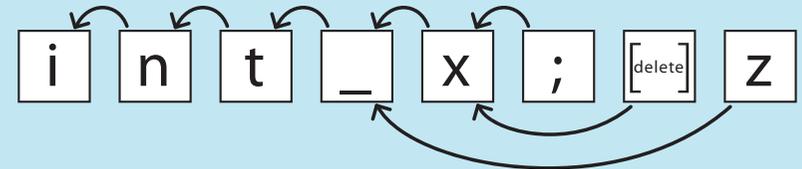
To record all code related data such as keystrokes, create files, copy and pastes, etc, and manipulate that data in order to tell a meaningful story.

### Ali Framework and Tools

#### Playback Events

By Time – Plays events back in the order they were created

By Position (Perfect Programmer) – Play events back in as if no mistakes were made allowing the user to ignore uninteresting history when creating a story



#### Create Stories

Create Important Clips – Brief playbacks that tell a specific story about a piece of code

Create Storyboards – Contain a collection of Clips that tell a story about the project

#### Time Frames (accurate to the second)

Entire duration of the Project – From project creation up to the most recent event

Within a Time Period – All events from a specified start date up to a specified end date

#### Grouping of Events

All Documents – All files within a project

A Single Document – A single specified file of a project

Selected Text – A portion of contiguous text and historical neighbors and parents of that text within a specified file of a project

#### Filtering of Events

All Events – All recorded events within a project

Events by Developer – All events created by a specified developer of the project

Events relating to a Feature – All events that have been tagged as corresponding to a specified feature of the project

## Future Plans

Use this software in existing IDE's such as Eclipse or MS Visual Studio

Integrate with existing Version Control Systems

Gather statistical information about programmers and how they write code

Extend to projects beyond Software such as Word Processing or Digital Art