

Data scientists 들은 보통 작은 code snippet 을 programming 하며, 그 code snippet 을 바꿔 나가면서, 그들의 idea 를 발전시킨다. 이를 exploratory programming 이라 한다. 그들은 code snippet 을 바꿀 때, Git 같은 VCS 를 쓰지 않고, informal 하게 comment out 한다거나, file 을 새로 만들거나 한다. 그 이유는 이렇게 하는 것이 훨씬 빠르고 쉽기 때문이고 file level 에서만 이루어지는 것이 아니기 때문 (comment out 의 경우) 이다. 그러나 이러한 informal 한 versioning method 를 사용할 경우, 프로그래머가 version 들의 strong mental map 을 가지고 있지 않으면, 어떤 version 이 어떠한 결과를 냈고, 어떤 idea 가 사용되었는지 혼란이 오게 되어서, 사용의 어려움을 겪게 한다. Variolite 는 이러한 문제를 해결하기 위해, local versioning 을 구현하였다. “variant boxes” 를 code 영역에 그리면, 그 안에 잇는 code 를 locally versioned 혹은 branched 할 수 있게 된다. 한 file 안에 여러 개 존재 할 수 있으며, nested 될 수도 있다. 또한 branch 혹은 commit 들에 이름을 달고, tag 를 붙이는 등 해당 commit 의 meta data 를 저장해 version 들을 구별 할 수 있게 해준다.

Usability 측면에서 보았을 때, learnability 와 memorability 가 굉장히 좋을 것이라고 판단하였다. Code snippet 을 variant box 로 만들고, branch 를 만든 후에, tab 을 선택하기만 하면 version 이 바뀌는 것이므로 기억하기도 쉽고 배우기도 쉬워 보인다. 실제 실험 결과도 10 명중 9 명이 성공적으로 code 를 variant box 로 만들었다. Effectiveness 측면의 경우 code snippet 들의 graph 가 유지되고 meta data 저장에 쉽다는 점에서 version 들의 mental map 을 유지 할 필요가 있는 informal versioning method 에 비해 좋아 보인다. Subjective satisfaction 의 경우 10 명의 실험자 모두 실제 삶에서 사용하고 싶다고 답한 점을 보아 좋아보인다.