

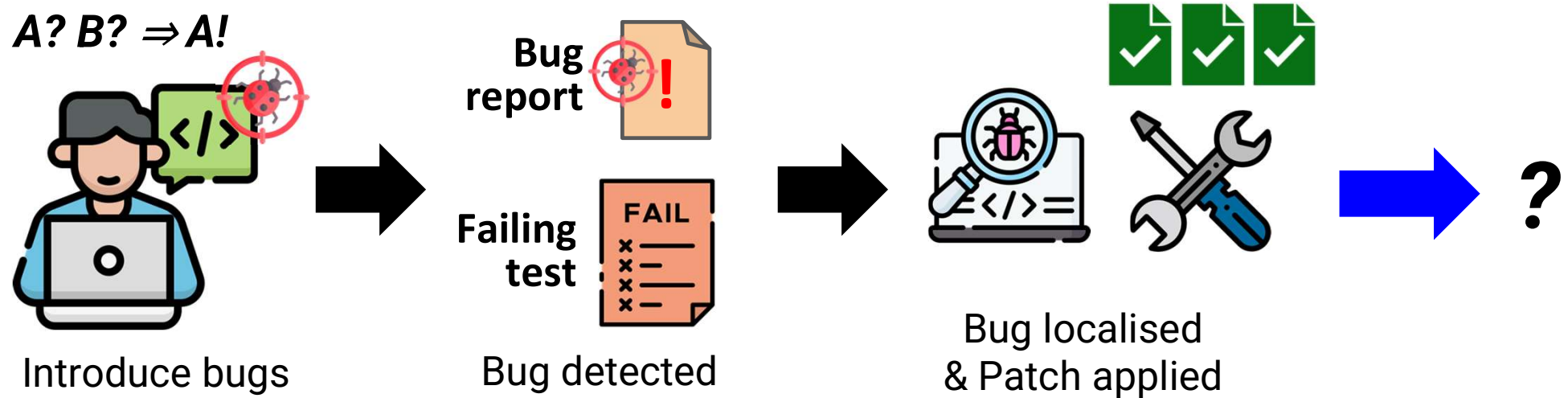
시간적 의미적 맥락을 활용한 소프트웨어 결함 및 불안전성 국소화

2026.02.03

경북대학교 손정주

2026 ERC 동계 워크숍

결함의 life-cycle: 결함 국소화 및 디버깅

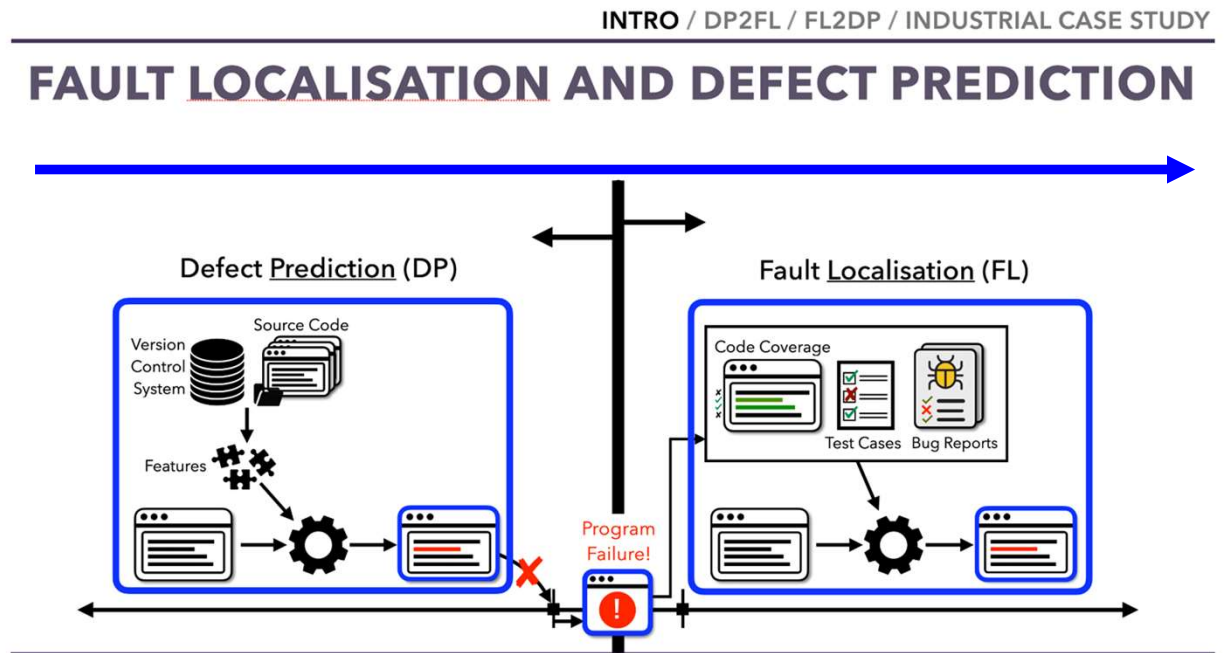


Q. 이 시점의 결함이 정말 문제의 '끝'인가?

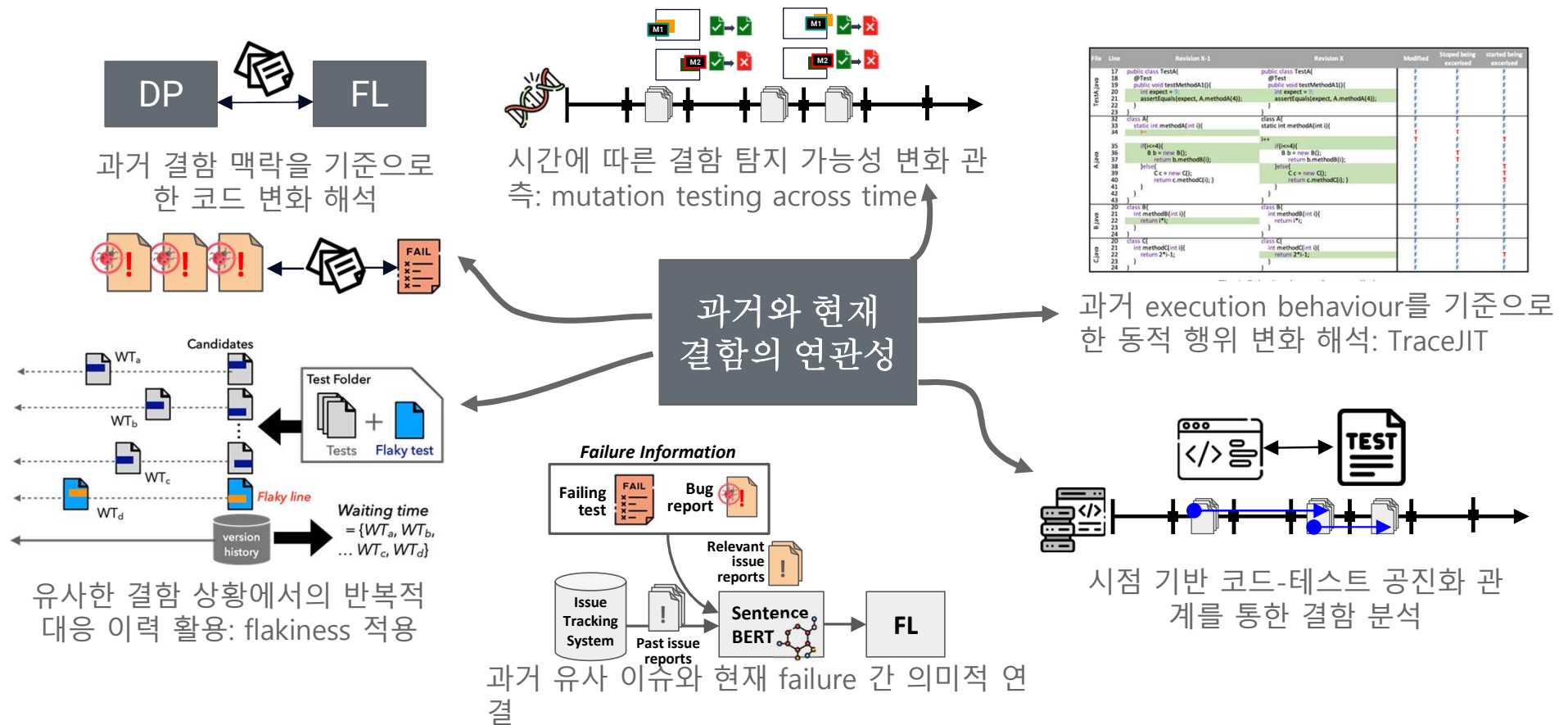
- 한동안 fault-free 한 상태로 유지되는가? 다음 결함은 언제, 어떤 형태로 드러나는? 그 결함은 이전에 발생한 문제와 무관한가?

이전 연구의 메인 테마: 과거와 현재 결함, 그리고 코드 변화

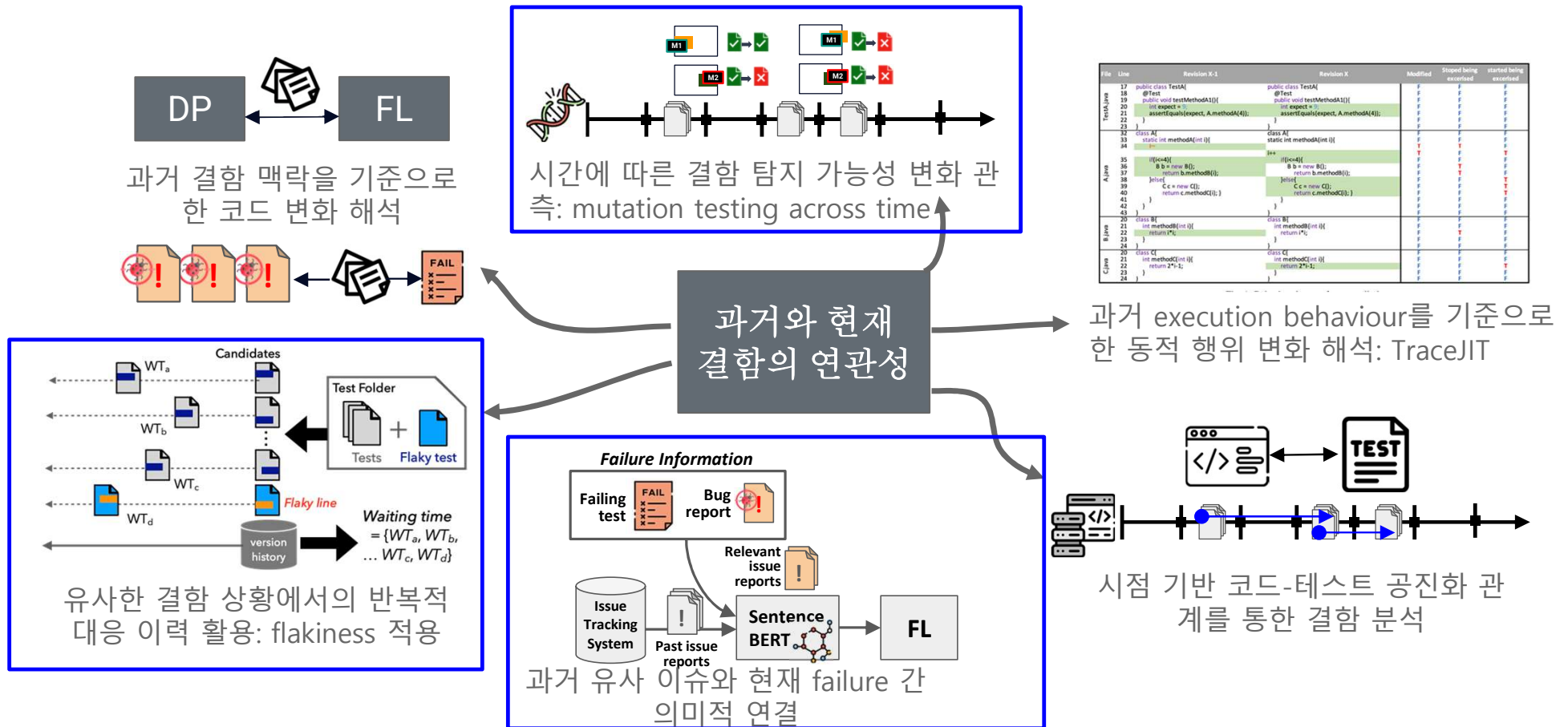
1. 과거와 현재의 결함은 서로 연관되어 있다.
2. 결함 예측과 결함 위치 식별은 결국 시점의 차이이다.
3. 개발자는 같은 실수를 반복한다.
4. 이들을 엮어보자!



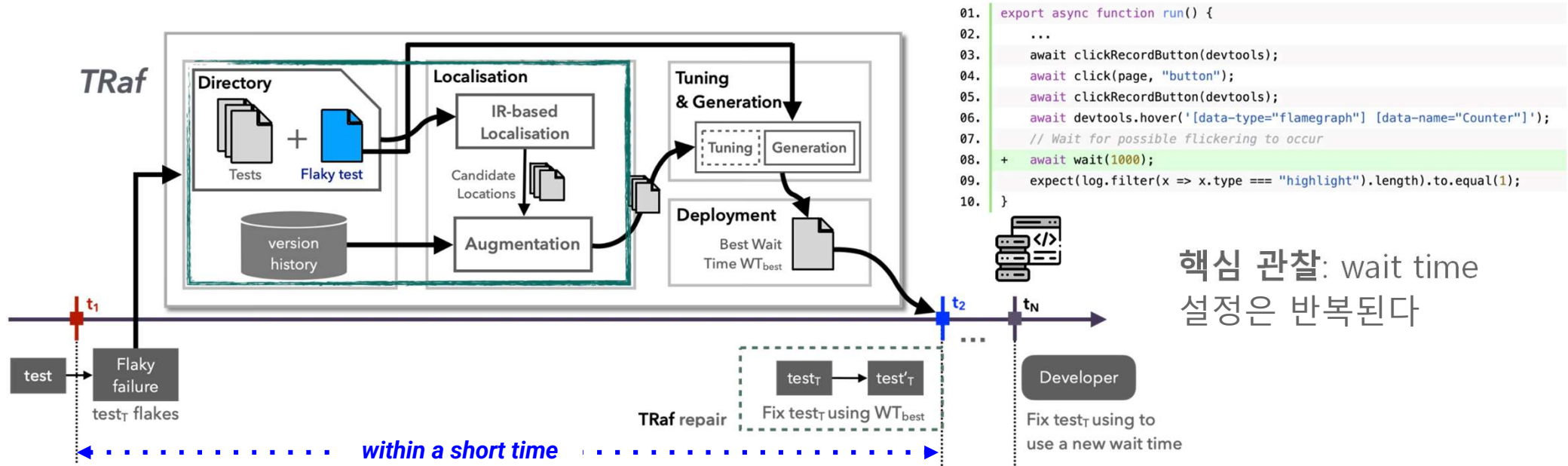
이전 연구의 메인 테마: 과거와 현재 결함, 그리고 코드 변화



이전 연구의 메인 테마: 과거와 현재 결함, 그리고 코드 변화

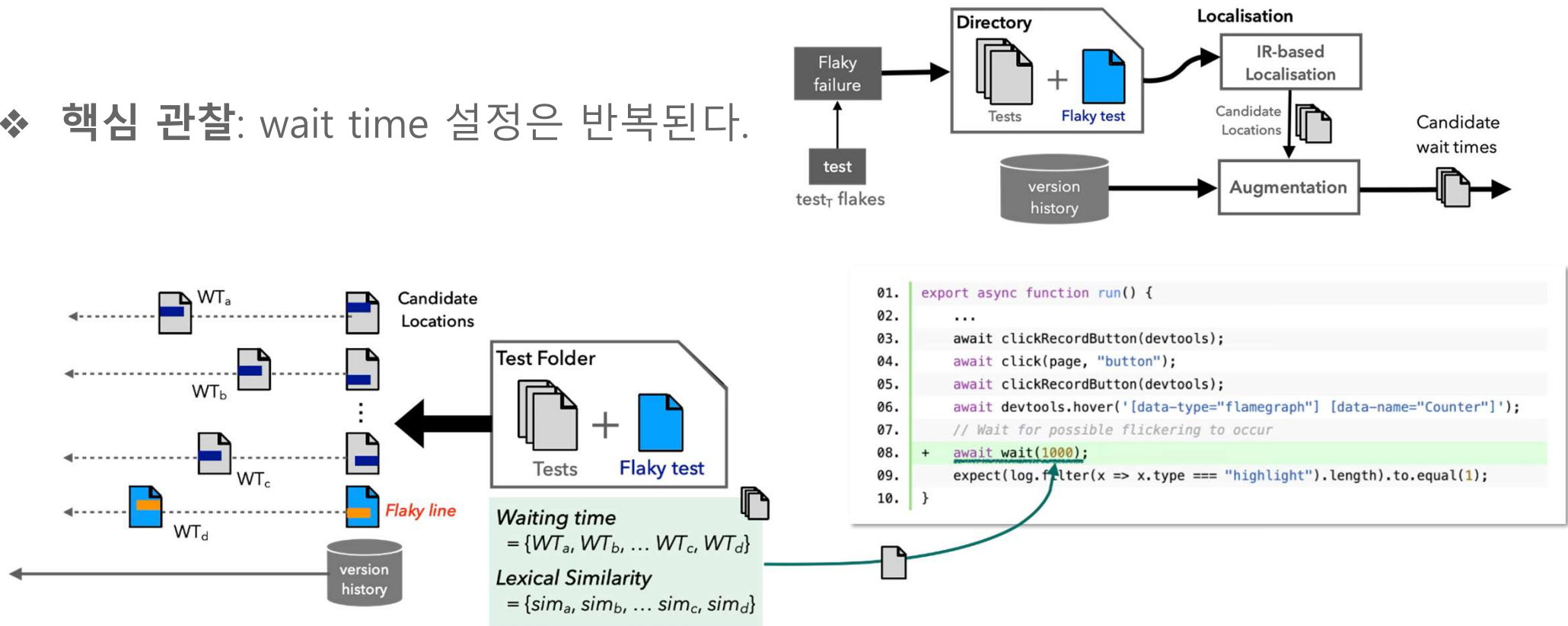


유사 코드 맥락에서의 시간 파라미터 재사용을 통한 불안정성 (flakiness) 완화



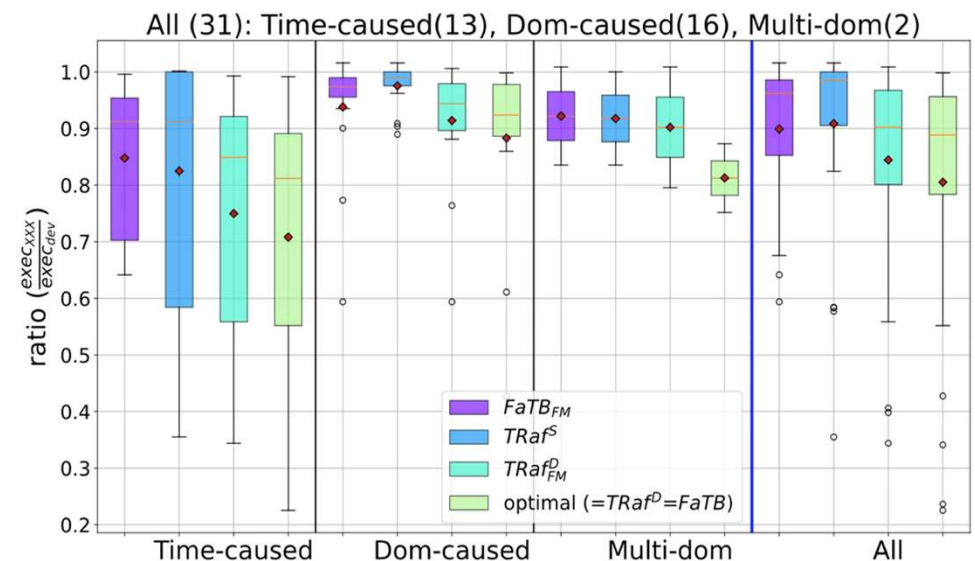
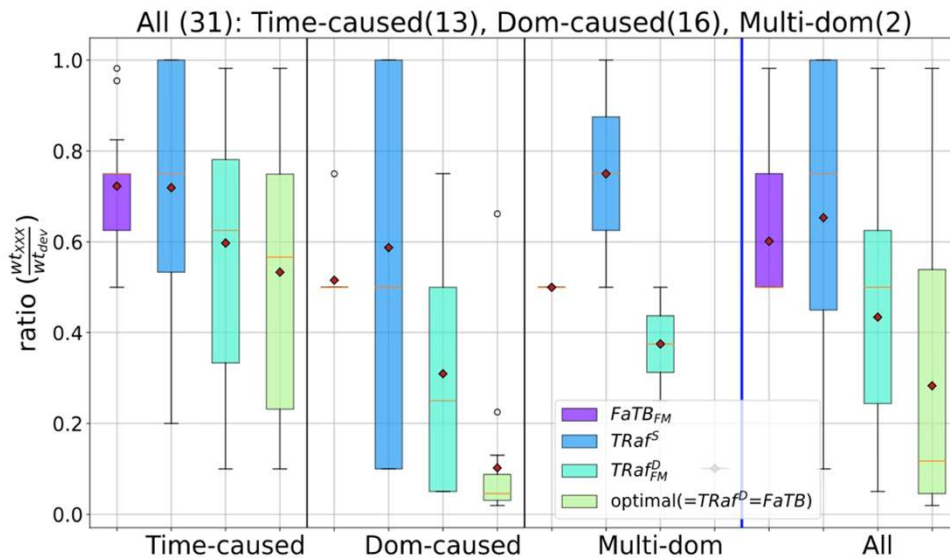
유사 코드 맥락에서의 시간 파라미터 재사용을 통한 불안정성 (flakiness) 완화

❖ 핵심 관찰: wait time 설정은 반복된다.



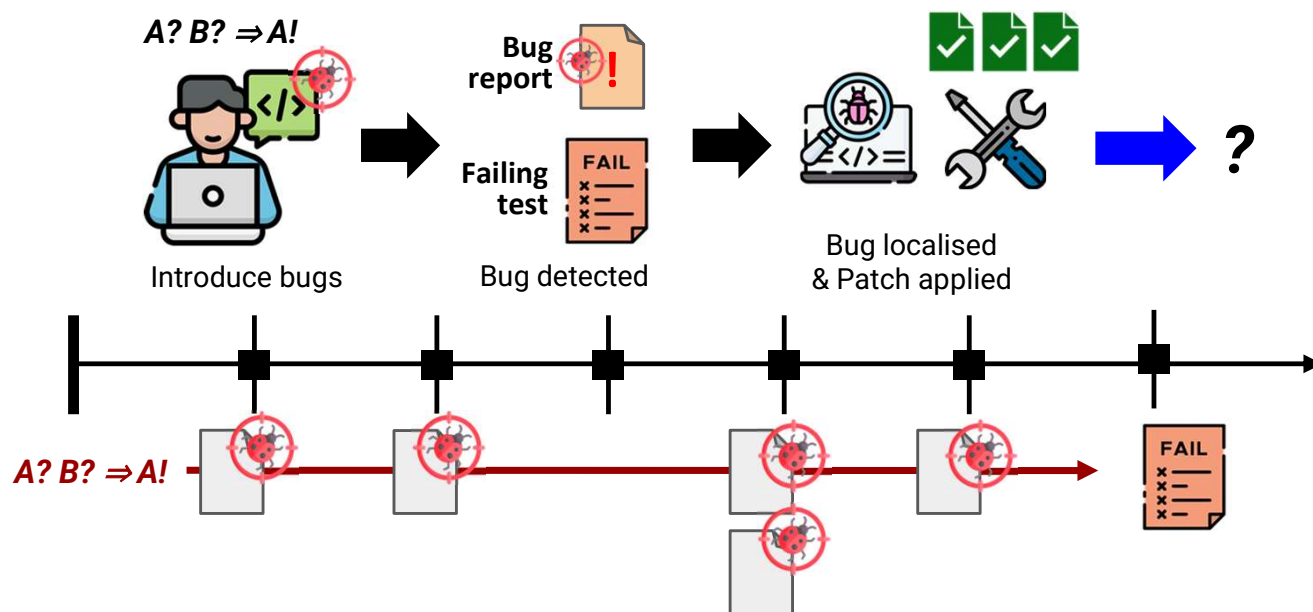
유사 코드 맥락에서의 시간 파라미터 재사용을 통한 불안전성 (flakiness) 완화

- 불안전성을 제거하면서도, 테스트 실행 시간 증가를 최소화하는 wait time 을 성공적으로 찾고 적용함



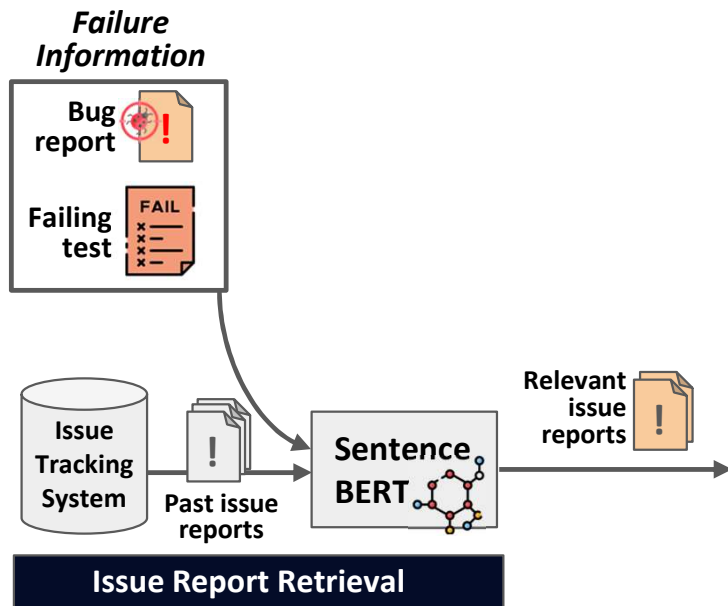
과거 유사 이슈와 현재 **failure**간 의미적 연결: ReportFL

❖ **핵심 관찰:** 개발자는 유사한 실수를 반복한다.



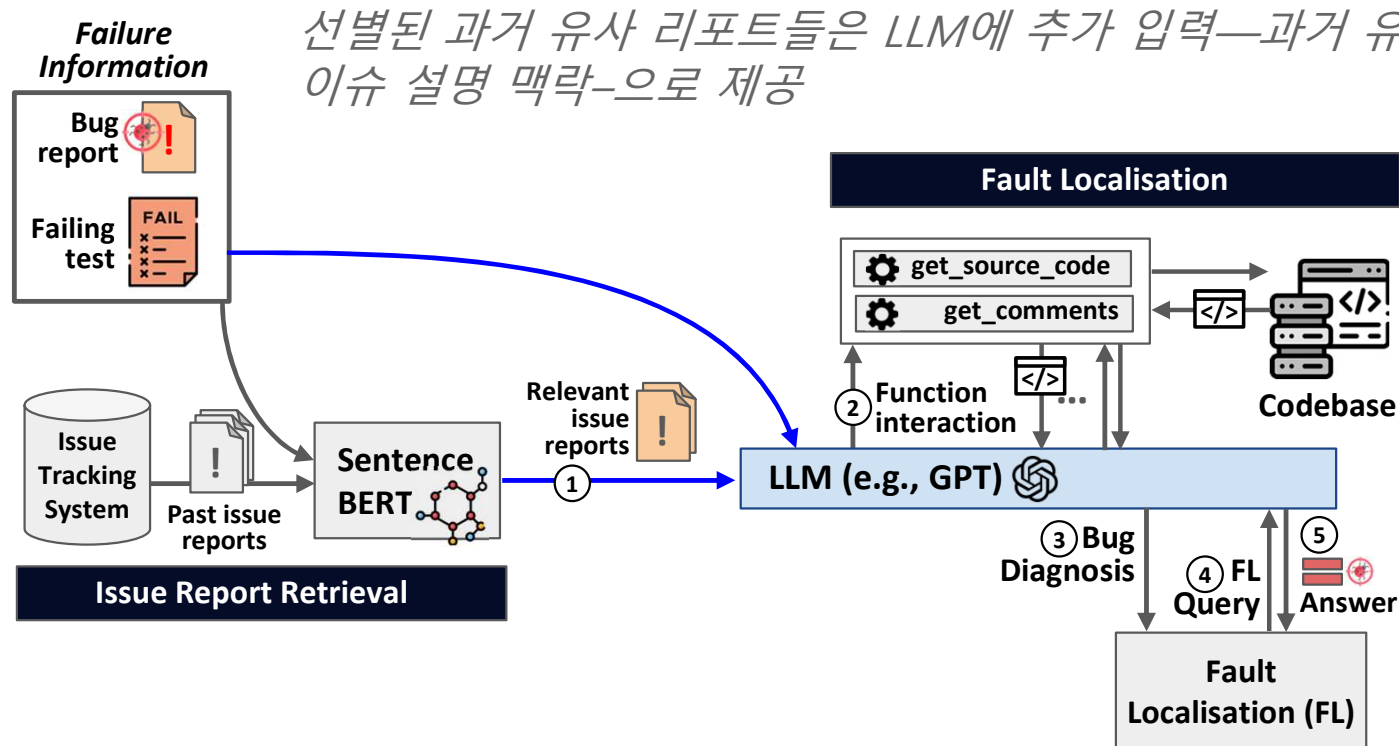
기존 접근: 현재 발생한 결함의
성질과 맥락에 초점 \Rightarrow 발생 맥
락을 과거 시점까지 확장해보자.

과거 유사 이슈와 현재 failure간 의미적 연결: ReportFL



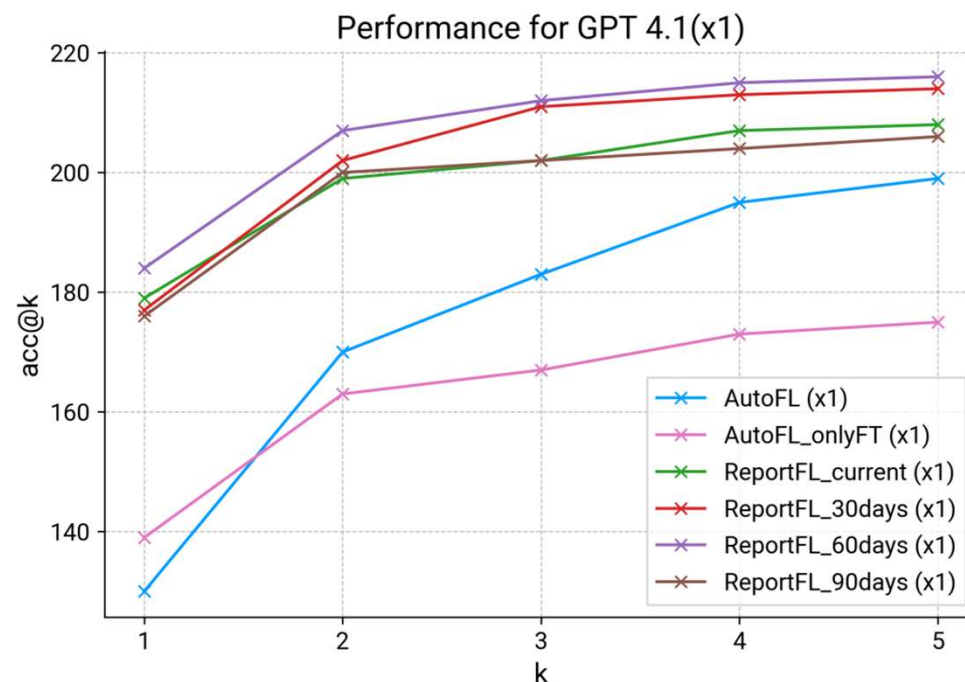
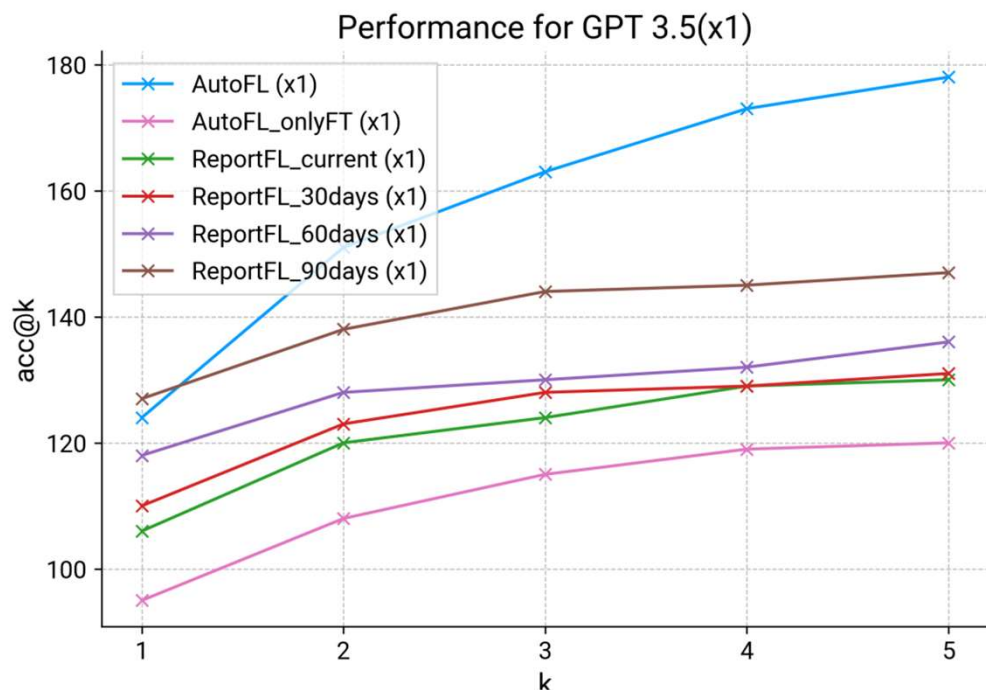
- **기존 접근:** 현재 발생한 결함의 성질과 맥락에 초점
⇒ 발생 맥락을 과거 시점의 "설명 맥락"까지 확장.
- **결함 및 이슈 리포트:** 결함 또는 이슈가 어떻게, 어떤 맥락에서 발생하였는지를 자연어로 기술.
 - 이슈 트래킹 시스템으로부터 과거 이슈 리포트 수집
 - 현재 Failure의 정보(버그 리포트 및 실패한 테스트)를 기반으로 과거 유사한 이슈의 설명 맥락을 활용.

과거 유사 이슈와 현재 failure간 의미적 연결: ReportFL



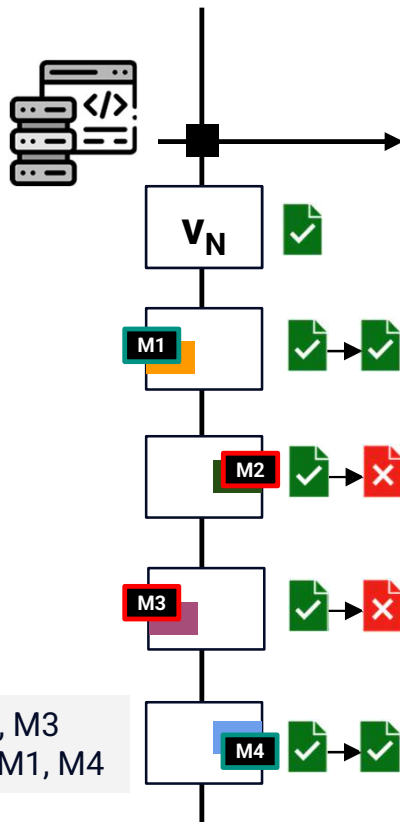
Q. 과거 이슈를 어느 시점까지 (time window) 확장해 활용 해야 하는가?

과거 유사 이슈와 현재 failure간 의미적 연결: ReportFL



Finding. 결함 위치 식별은 현재 failure에 국한되지 않으며, 과거 결함과 이슈의 맥락과 연속선적으로 연결되어 있다.

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time

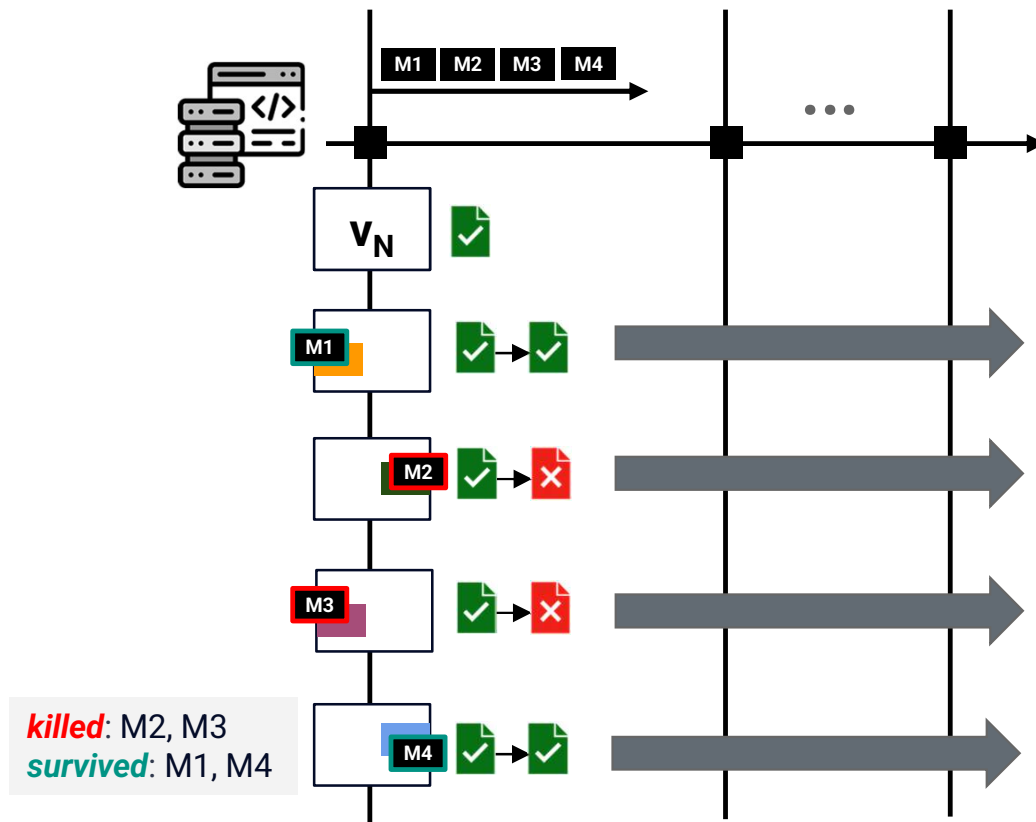


- **뮤테이션 테스트:** 코드에 인위적 결함인 뮤턴트를 삽입, 테스트가 결함을 얼마나 잘 드러내는지(kill)를 관측
- 기존의 뮤테이션 테스트는,
 - 하나의 시점에서 뮤턴트의 탐지 여부 평가
 - 현 시점의 test suite 품질 평가 및 결함 식별 (MBFL)

“결함은 등장과 동시에 항상 탐지 가능한 상태로 드러나지 않는다.” \Rightarrow 결함을 드러내는 “신호”의 가시성은 시간에 따라 달라질 수 있다.

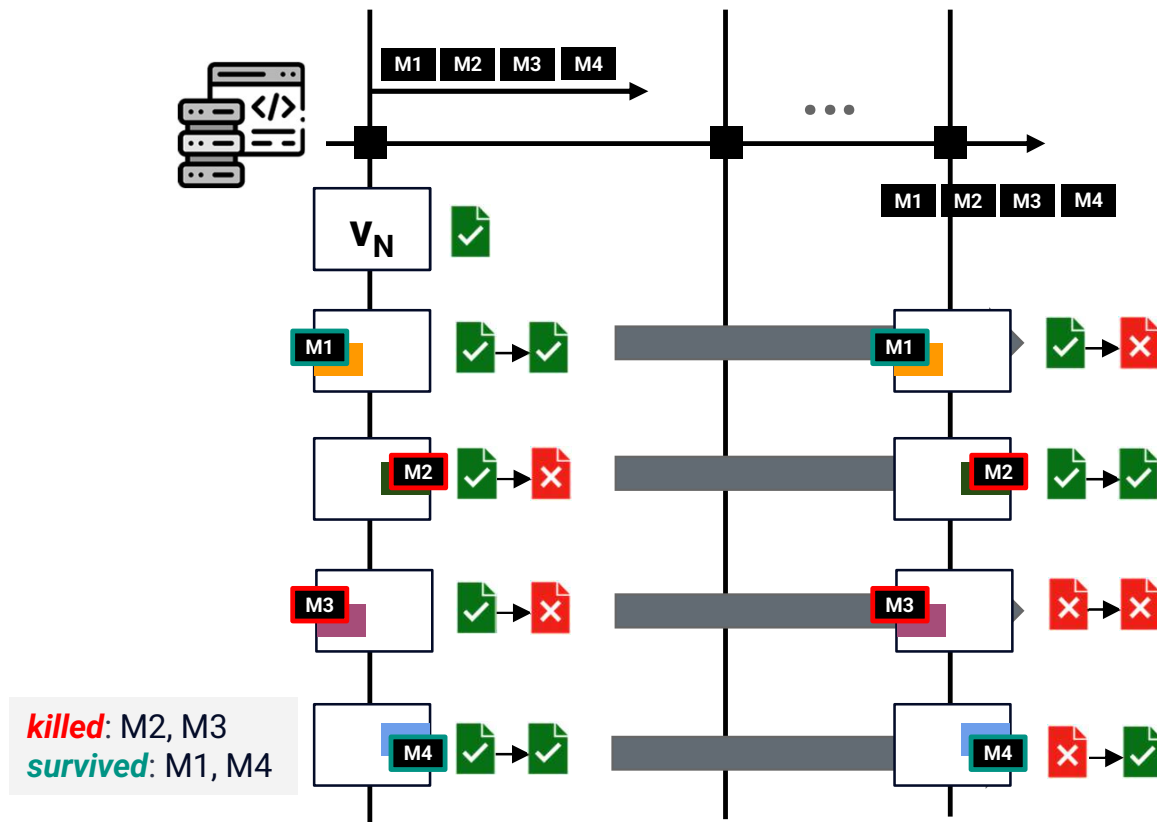
Q. 코드와 테스트가 진화하는 상황에서도, 뮤턴트 탐지 가능성은 항상 동일할까?

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time



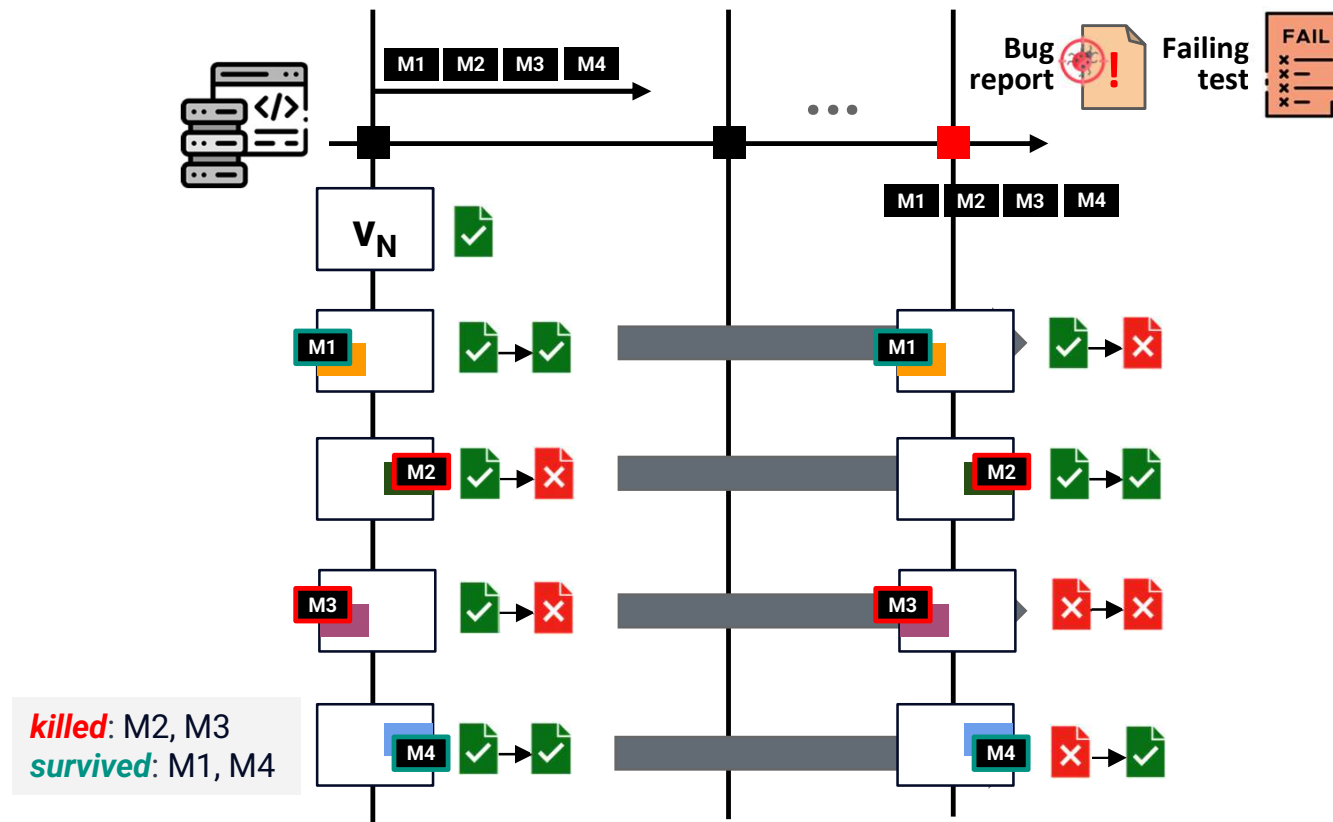
동일한 뮤텐트를 코드 및 테스트 변화 속으로 전파하며, 시간에 따른 테스트 탐지 (kill) 여부를 기록 및 분석

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time



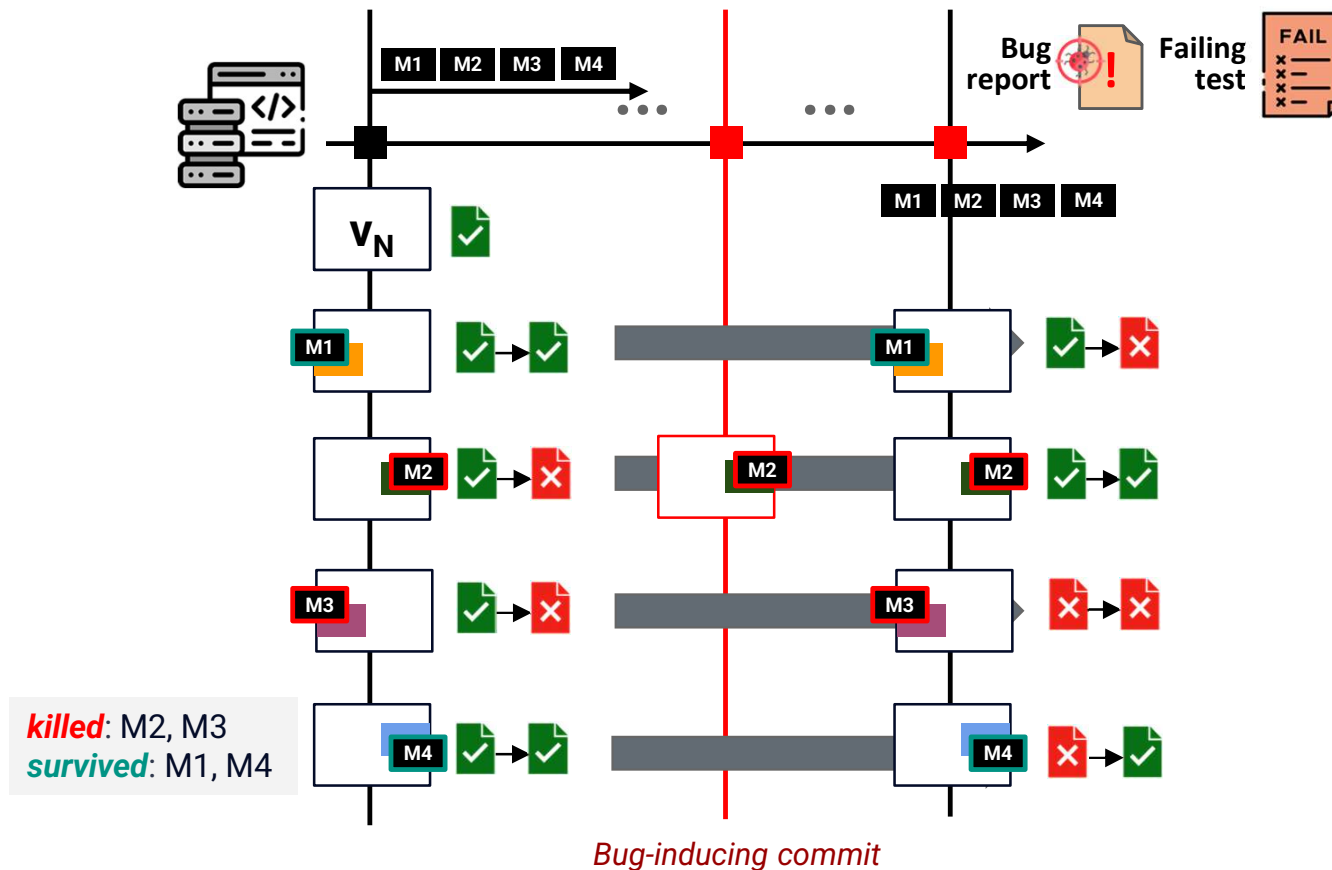
동일한 뮤턴트를 코드 및 테스트 변화 속으로 전파하며, 시간에 따른 테스트 탐지(kill) 여부를 기록 및 분석

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time



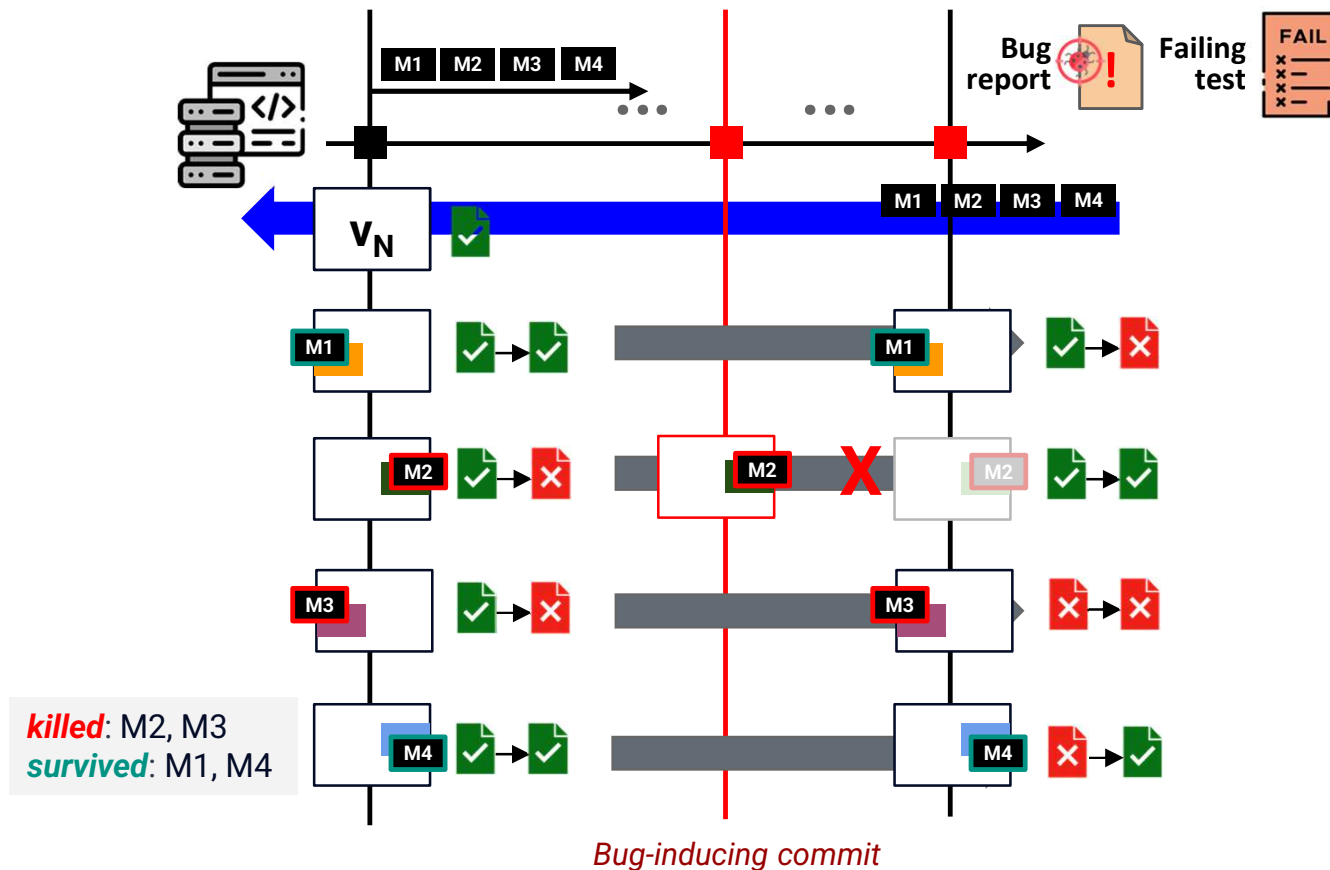
Q. 어떤 뮤턴트가 결함과 연관되었는가 (coupled)?

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time



Q. 어떤 뮤턴트가 결함과 연관되었는가 (coupled)?

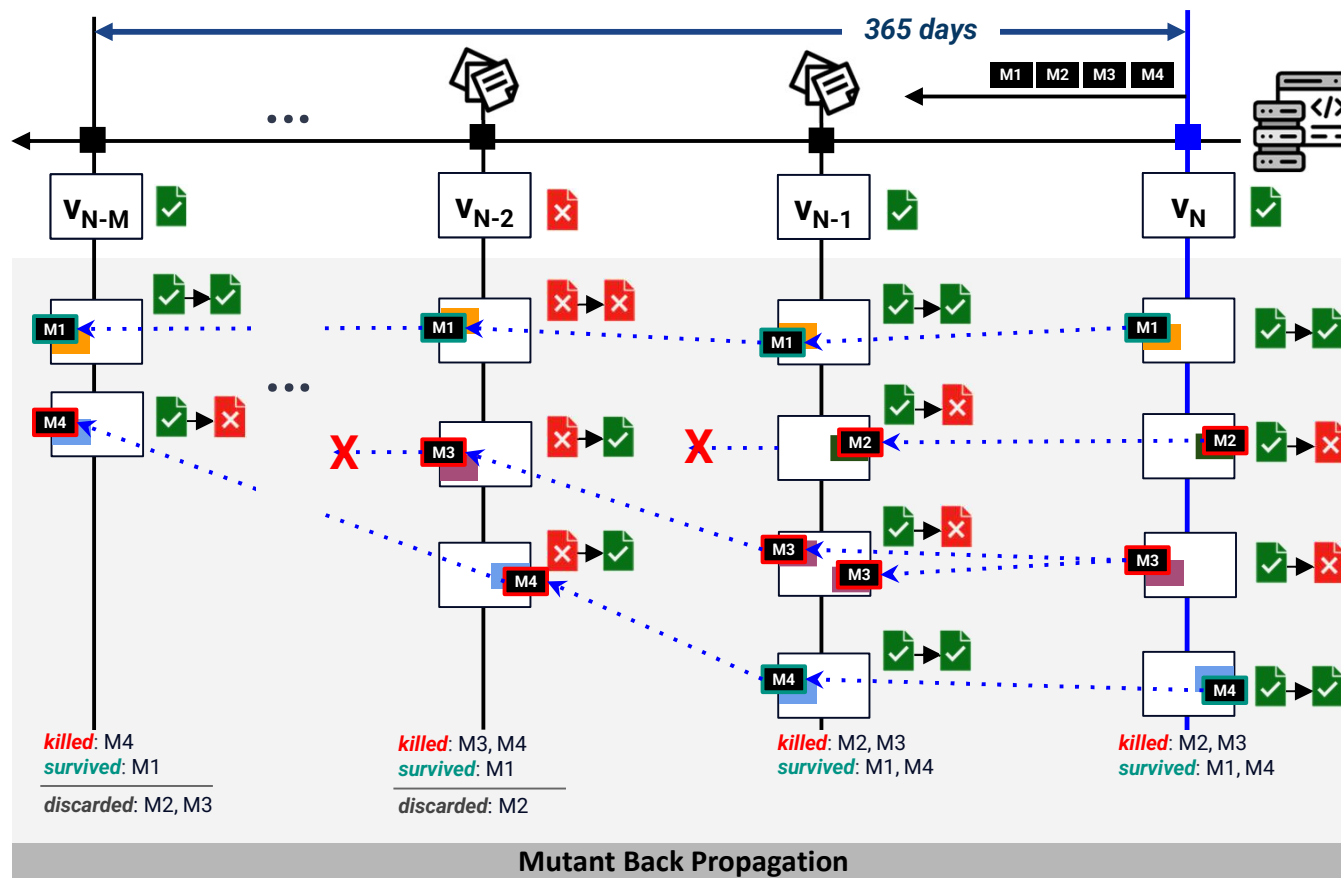
시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time



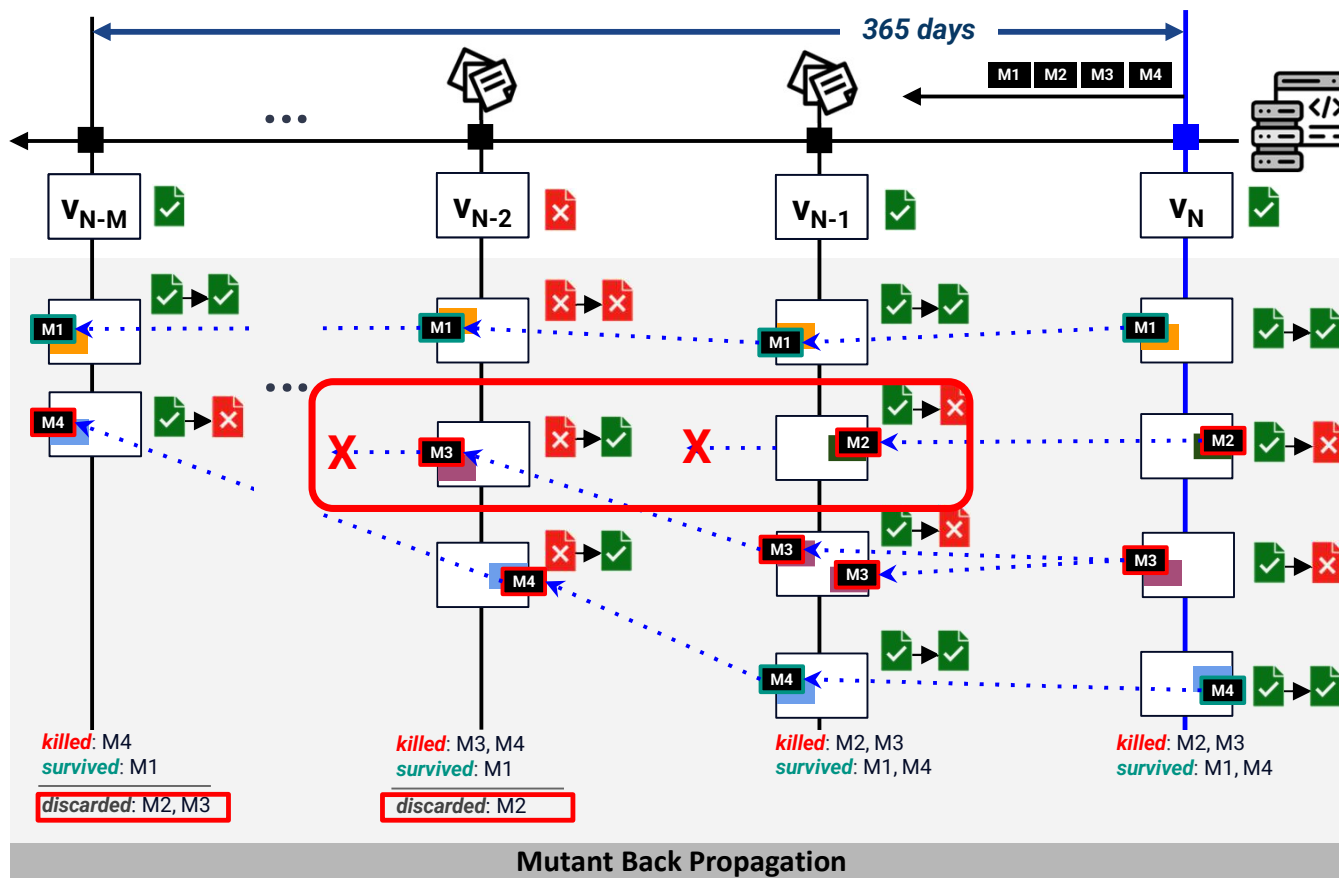
Q. 어떤 뮤턴트가 결함과 연관되었는가 (coupled)?

⇒ **Mutant Back-Propagation**

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time



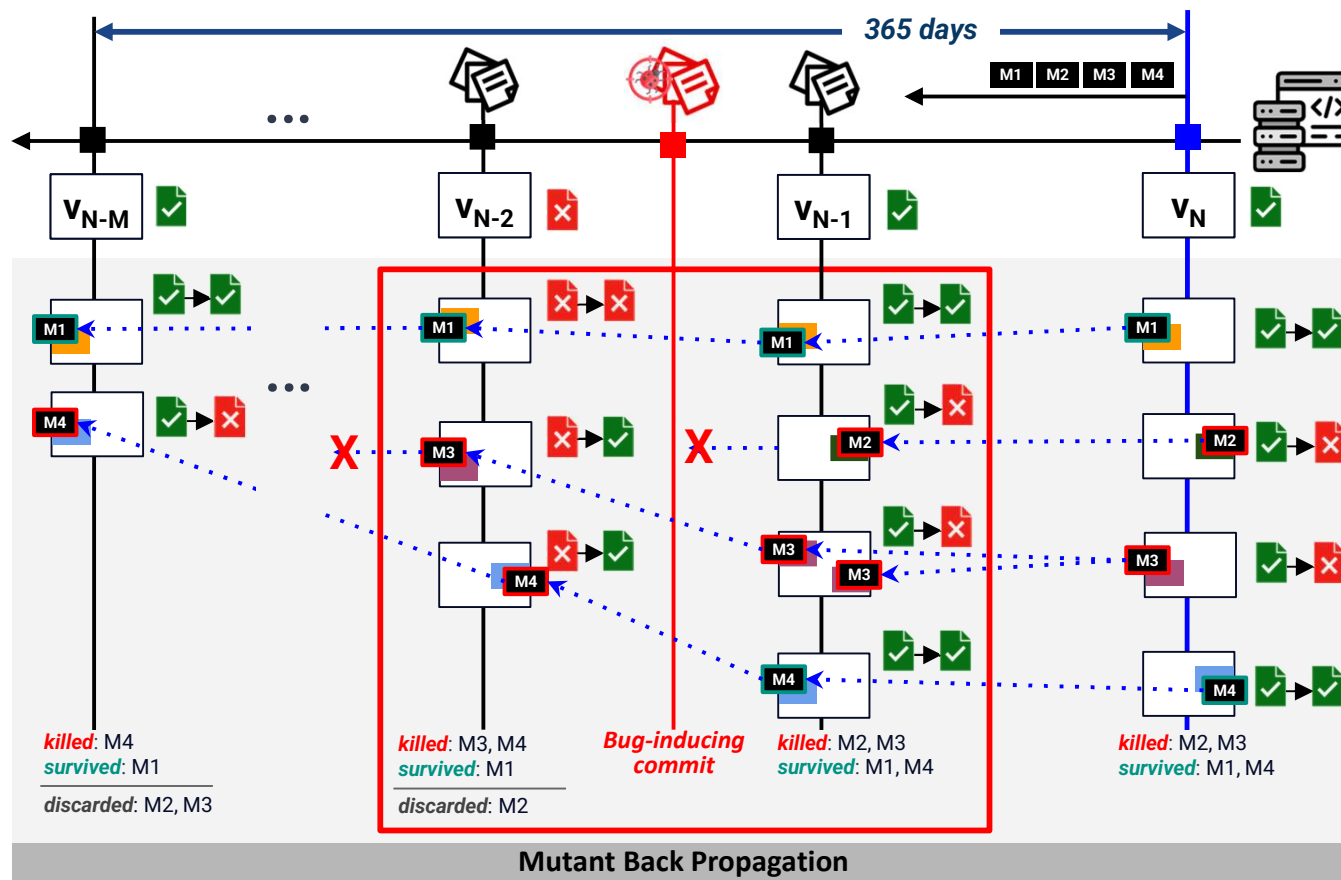
시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time



Discarded mutant:

중간 mutated 코드 요소가
삭제(새로 등장한 요소)되거나
코드 변화 추적에 실패.

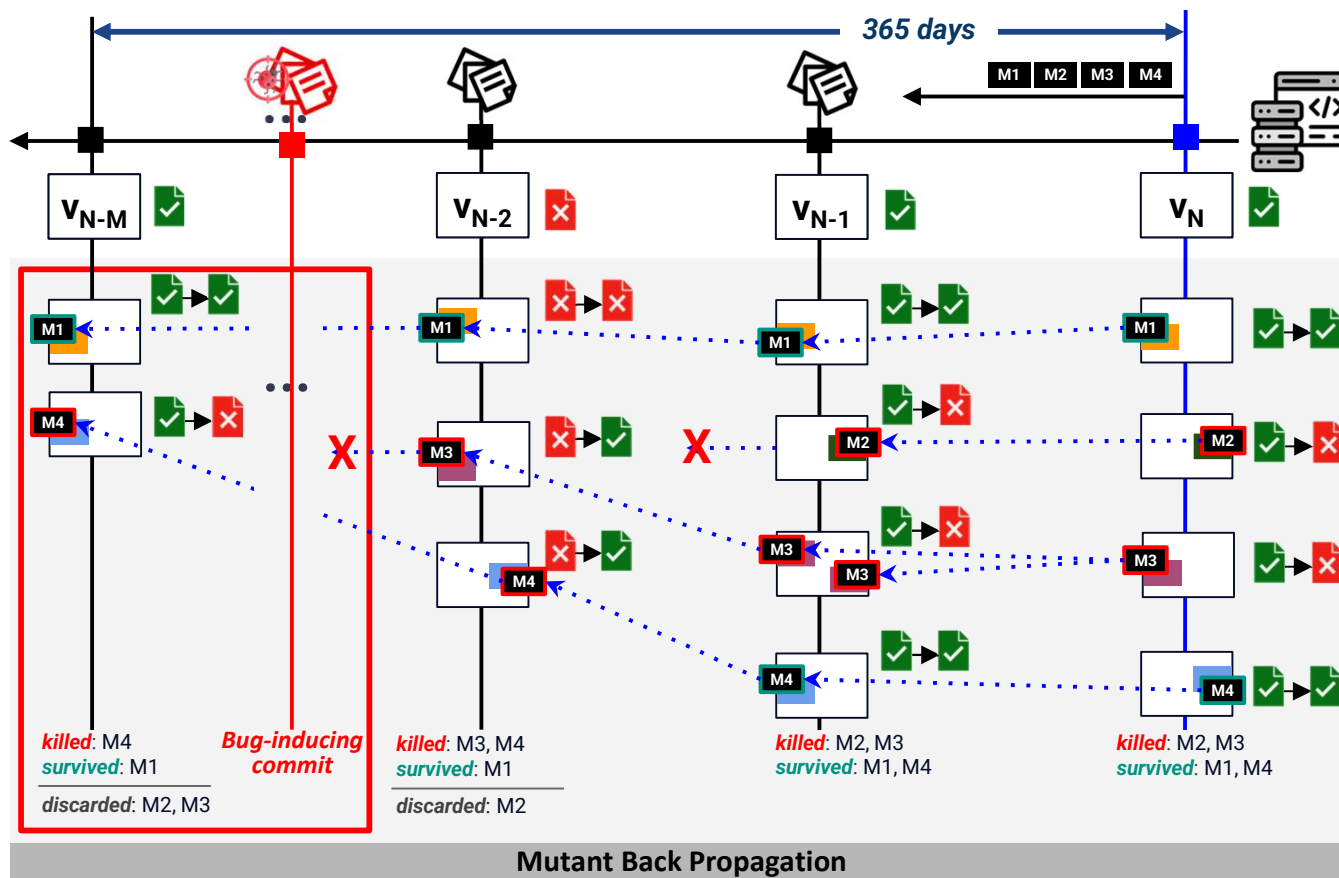
시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time



Discarded mutant:

중간 mutated 코드 요소가 삭제(새로 등장한 요소)되거나 코드 변화 추적에 실패.

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time

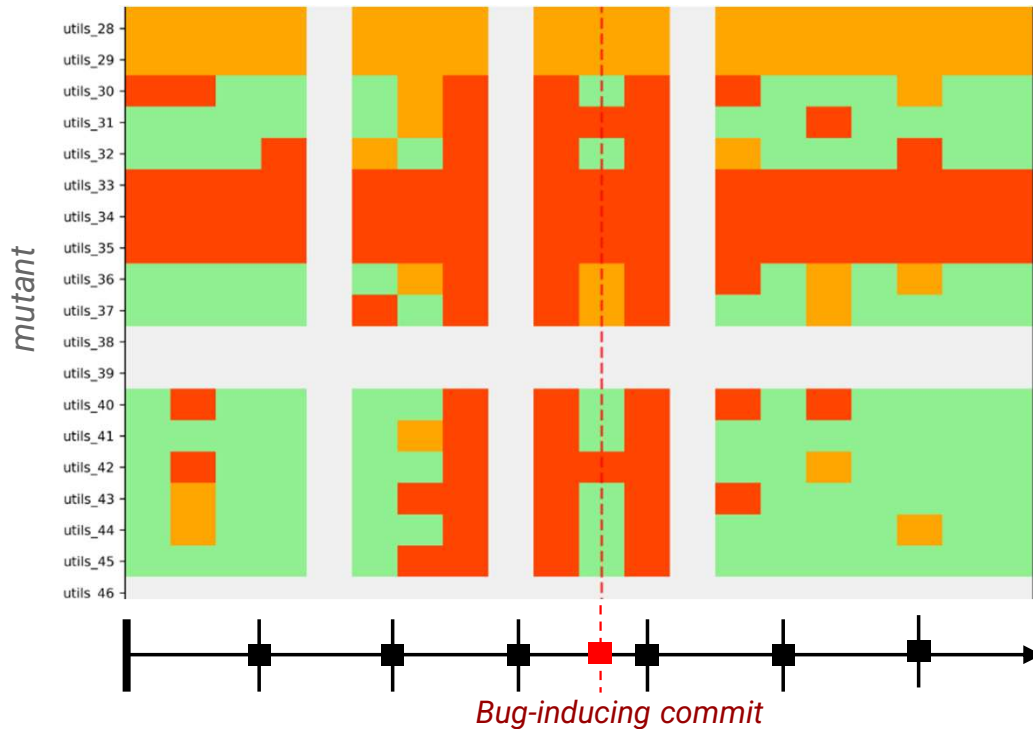


Discarded mutant:

중간 mutated 코드 요소가 삭제(새로 등장한 요소)되거나 코드 변화 트래킹에 실패.

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time

타겟 결함: thefuck-12b



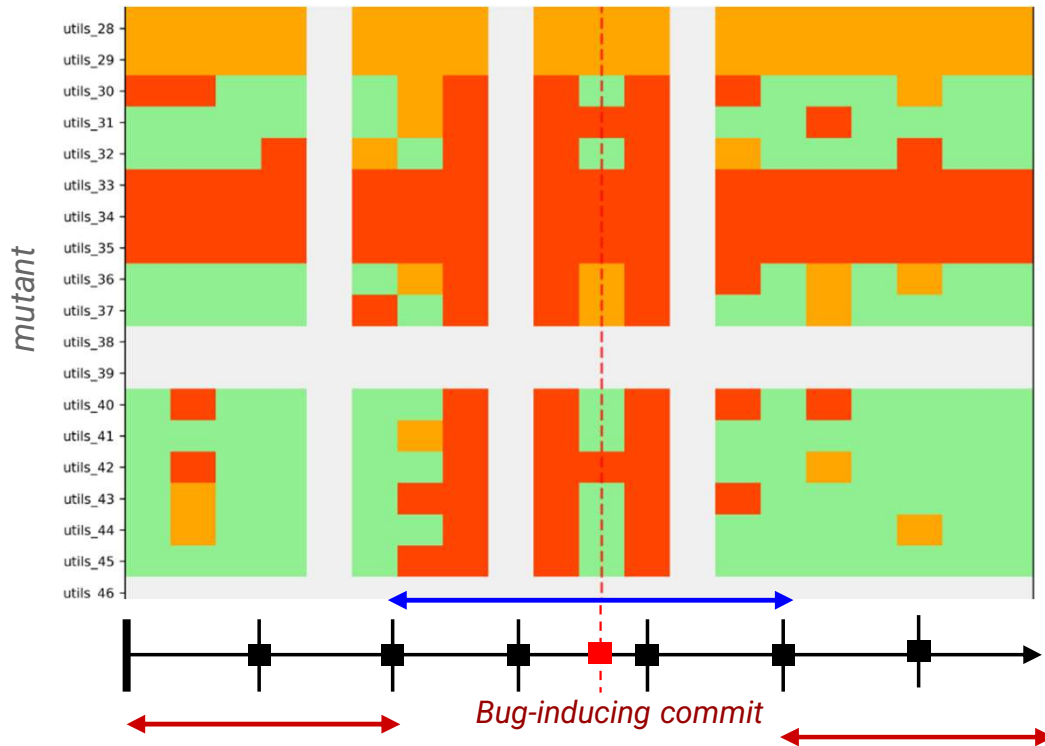
뮤턴트 관측 상태:

- **Stable:** 모니터링 기간 동안 동일한 탐지 상태 유지-계속 killed 또는 survived
- **Monotonic:** 모니터링 기간 동안 탐지 상태가 한번 전환
- **Oscillatory:** 모니터링 기간 동안 탐지 상태가 두번 이상 전환

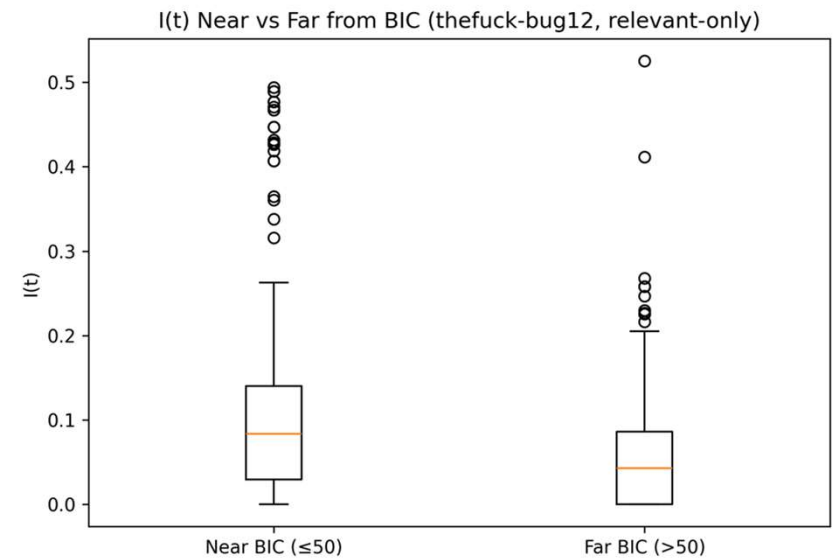
project_name	stable	monotonic	oscillatory	unknown
cookiecutter-2b	189	17	5	0
fastapi-2b	264	0	0	294
httpie-2b	329	1	1	0
sanic-1b	252	29	7	405
thefuck-12b	52	5	144	1
thefuck-13b	40	13	158	1
thefuck-14b	85	9	187	1
thefuck-17b	111	26	205	1
thefuck-18b	212	24	106	0

시간에 따른 결함 탐지 가능성 변화 관측: mutation testing across time

타겟 결함: thefuck-12b

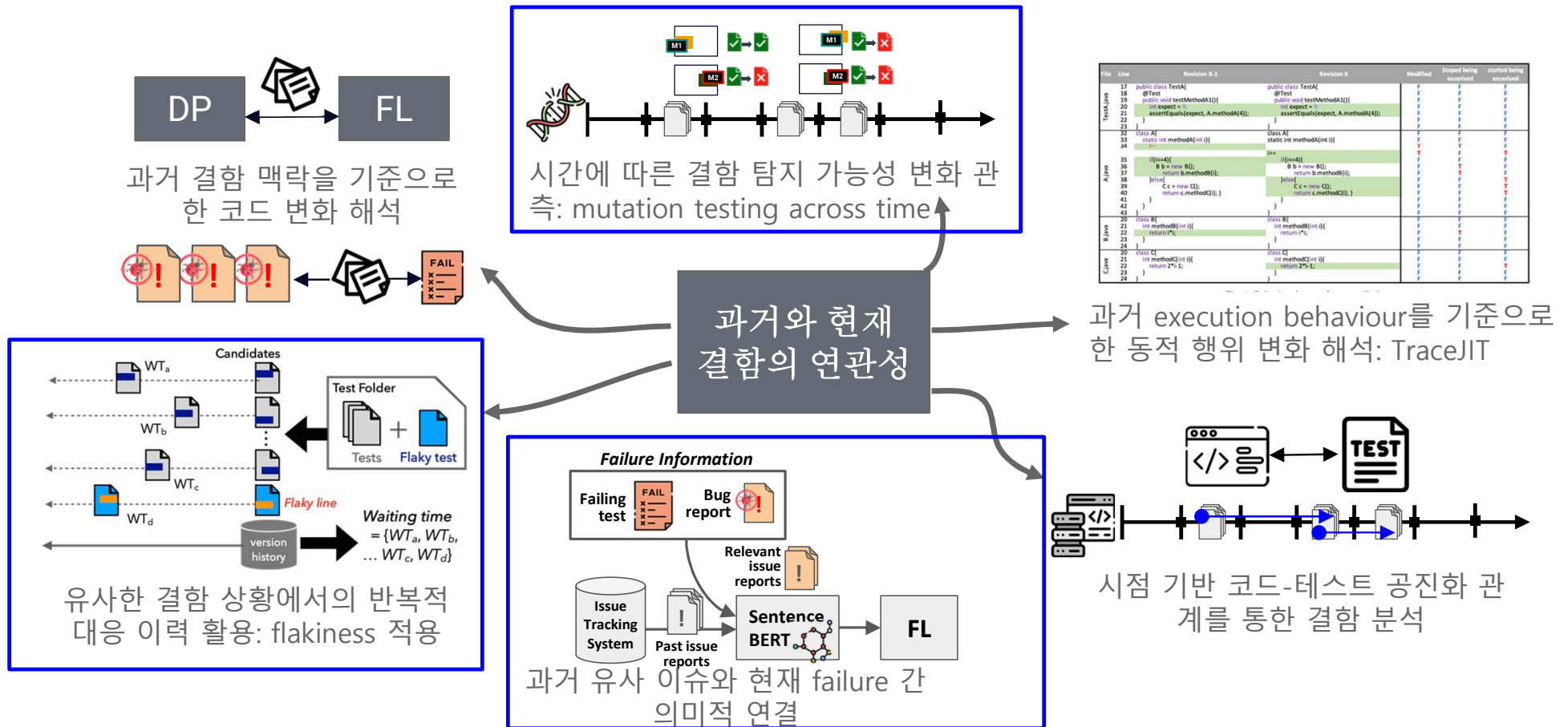


$$I(t) = \frac{|\{m | status_m(t-1) \neq status_m(t)\}|}{|\{m | status_m(t-1), status_m(t) \text{ both observed}\}|}$$



결함 유입 시점에서의 상대적 불안전성
⇒ 새로운 결함 유입 시그널

이전 연구의 메인 테마: 과거와 현재 결함, 그리고 코드 변화



결론 및 향후 연구 방향

결함은 단일 시점의 실패로 이해될 수 없으며, 과거와 시간에 걸친 코드, 테스트, 관측 시그널의 연속 속에서 이해되어야 한다.