

회고

1. 업무 효율성 향상
 1. **Playwright Codegen** 레코딩 툴을 활용하여 테스트 스크립트 자동 생성
 2. **AI 도구 활용**
 1. 요구사항 정리 후 **ChatGPT**를 통해 테스트 시나리오 및 케이스 도출
 2. 기능 별 자동화 스크립트 초안 작성
 3. **API** 테스트 시 커버리지 조합
2. 크로스 브라우징 정책 검토
 1. 테스트 대상 브라우저: **Chrome, Edge, Firefox**
3. 회원가입 시 **SMS** 본인인증 대응 방안
 1. 사내 테스트 시 본인인증 우회 처리 구현
 2. 콘솔창에 인증번호 노출되는 이스터에그 기능 구현 (**JavaScript** 처리)
 3. **SMS API**를 통해 인증번호 직접 획득
4. 소셜 로그인 시 캡차 대응
 1. 카카오/네이버 계정 반복 로그인 시 캡차 노출
 2. 테스트 환경에서는 캡차 회피 로직 또는 수동 인증 처리 필요
5. 자동화 스크립트 작성 시 고려 사항
 1. 회원가입 시 테스트 계정 누적 → 정보보호 이슈 발생 가능
 2. 결제 시나리오
 1. 결제 후 취소 처리 포함 필요
 2. 네이버페이: 키패드 동적 변경 발생 → **OpenCV**로 대응 필요
6. 테스트 자동화 실행 시간 협의
 1. 수동 수행
 2. **Jenkins** 예약 실행 (예: 새벽 3시)
 3. **Jenkins Trigger**를 이용한 배포 후 자동 실행
7. 테스트 결과 공유
 1. **Slack** 연동을 통한 테스트 결과 알림
 2. **Jira, TestRail** 등 테스트 툴과 **API** 연동
8. 요구사항 정리 및 리스크 분석
 1. 요구사항 도출 후 개발자와 리스크 영역 사전 협의
 2. 리스크 기반 테스트 설계 및 검수 일정 산정
9. 서버 환경 확인 필요
 1. 테스트 환경 구분: **QA / STG / 운영**
 2. 환경별 테스트 시나리오 적용 여부 검토
10. 자동화 스크립트 재활용 범위
 1. 공통 시나리오 모듈화 (**POM** 구조 적용)
 2. 우선순위 등급 별 범위 논의 필요
11. 수동 테스트 vs 자동 테스트 비율 조정
 1. 테스트 전략에 따라 비율 유연하게 조정