



프로젝트 종합 평가 보고서

이 문서는 Vibe Coding Report VS Code 확장에서 수집한 스냅샷과 세션 데이터를 기반으로, 현재 프로젝트 상태를 정리한 평가 문서입니다.

devplan/Session_History.md 파일에는 개별 세션별 상세 로그가 별도로 관리됩니다.

🎯 프로젝트 목표 및 비전

• 프로젝트 목적

- VS Code에서 AI 페어 프로그래밍을 사용할 때, 프로젝트 구조와 변경 이력을 자동으로 분석하여
 - 종합 평가 보고서(Project Evaluation Report)
 - 개선 탐색 보고서(Project Improvement Exploration Report)
 - AI 실행용 [Prompt.md](#)를 한 번의 명령으로 생성·유지관리하는 도구입니다.

• 핵심 목표

- 워크스페이스를 자동 스캔하여 언어/구조/설정 정보를 수집
- Git 변경 이력과 결합한 증분 분석(Incremental Update) 제공
- AI 모델이 바로 사용할 수 있는 구조화된 프롬프트를 자동 생성
- 이미 적용된 개선 항목을 추적하여 중복 제안을 줄이고, 세션 히스토리를 시각적으로 관리

• 대상 사용자

- GitHub Copilot Chat 등 AI 도구를 활용해 프로젝트를 설계·리팩토링·문서화하는 VS Code 사용자
- 팀/개인 프로젝트에서 “현재 상태 파악 → 개선 항목 도출 → AI에게 실행 의뢰” 흐름을 반복적으로 사용하는 개발자



TL;DR (Executive Summary)

항목	내용
현재 버전	v0.3.11 (2025-12-04)
종합 점수	88점 (B+등급)
핵심 성과	명령 레이어 테스트 16개 추가 (102개 총), snapshotCache/markerUtils 모듈 분리

항목	내용
주요 위험	AI 직접 연동 미구현 (Low)
즉시 필요 조치	P3: AI 직접 통합 (feat-ai-integration-001)
다음 단계	P3: 멀티 워크스페이스 지원

⚠ Risk Summary

Risk ID	영역	위험 수준	설명	완화 전략
RISK-001	Integration	● Low	AI API 직접 통합 없음 - 현재 클립보드 방식에 의존	feat-ai-integration-001 구현으로 선택적 AI 통합
RISK-002	Scalability	● Low	단일 워크스페이스만 지원 - 멀티루트 워크스페이스 제한적	feat-multi-workspace-001 구현으로 해결

📋 프로젝트 개요

항목	값
프로젝트명	projectmanager (Vibe Coding Report 확장)
버전	0.3.11
최초 분석일	2025-11-29 15:48
최근 분석일	2025-12-04 23:10
파일 수	60
디렉토리 수	16
주요 언어	TypeScript(34), Markdown(8), JSON(4), YAML(1), YML(1)
프레임워크	VS Code Extension (Node.js / TypeScript)
테스트	102개 통과 (Vitest)

항목	값
이번 분석 메모	명령 레이어 테스트 추가(16개), snapshotCache.ts 및 markerUtils.ts 신규 모듈 추가

현재 구현된 기능

기능	상태	설명	평가
삼중 보고서 시스템 (Evaluation/Improvement/Prompt)	 완료	devplan 디렉토리에 평가·개선·프롬프트 파일을 생성하고, 마커 기반으로 섹션별 갱신을 수행합니다.	 우수
워크스페이스 스캔 및 스냅샷 수집	 완료	WorkspaceScanner가 언어 통계, 주요 설정 파일, 디렉토리 구조, Git 정보(옵션)를 수집해 ProjectSnapshot을 구성합니다.	 우수
Git 기반 변경 분석 (diff)	 완료	SnapshotService가 이전 스냅샷과 비교하여 새 파일/삭제 파일/설정 변경/Git 변경 목록을 요약합니다.	 우수
AI 연동을 통한 보고서 업데이트	 부분	프롬프트를 생성해 클립보드에 복사하고 Copilot Chat을 자동으로 열어주지만, Language Model API를 통한 직접 호출은 아직 미구현입니다.	 양호
개선 항목 추출 및 미적용 필터링	 완료	마크다운에서 P1/P2/P3 개선 항목을 파싱하고, appliedImprovements 기반으로 적용된 항목을 제외합니다.	 우수
세션 히스토리 및 통계 관리	 완료	.vscode/vibereport-state.json과 Session_History.md에 세션 목록과 통계를 기록하고, 사이드바 뷰에서 시각화합니다.	 우수
VS Code 사이드바 Summary/History/Settings 뷰	 완료	Summary(요약) Webview, History TreeView, Settings Webview를 통해 보고서 상태와 설정을 한 곳에서 관리합니다.	 우수

기능	상태	설명	평가
개선 항목 프롬프트 생성(Generate Prompt)	<input checked="" type="checkbox"/> 완료	개선 보고서에서 미적용 항목을 QuickPick UI로 선택해 Prompt.md를 생성하고 클립보드에 복사합니다.	● 우수
프로젝트 비전(Project Vision) 설정	<input checked="" type="checkbox"/> 완료	QuickPick/Input UI로 Project Vision을 설정하고, Settings 패널에서 직접 모드/유형/단계를 변경할 수 있습니다.	● 우수
테스트 및 CI 파이프라인	<input checked="" type="checkbox"/> 완료	Vitest 기반 단위 테스트 74개와 GitHub Actions CI 워크플로우가 구성되어 있습니다.	● 우수
점수-등급 일관성 시스템	<input checked="" type="checkbox"/> 완료	SCORE_GRADE_CRITERIA 상수와 scoreToGrade/gradeToColor 헬퍼 함수로 일관된 평가를 보장합니다.	● 우수
파트별 순차 작성 지침	<input checked="" type="checkbox"/> 완료	AI 에이전트 출력 길이 제한 방지를 위한 파트별 분리 작성 가이드라인을 제공합니다.	● 우수



종합 점수 요약

아래 점수는 v0.3.11 기준 11차 평가 결과입니다.

명령 레이어 테스트 16개 추가, snapshotCache/markerUtils 모듈 분리로 품질 지표가 향상되었습니다.

항목	점수 (100점 만점)	등급	변화
코드 품질	92	● A-	▲ +1
아키텍처 설계	88	● B+	▲ +2
보안	73	● C	-
성능	80	● B-	▲ +1
테스트 커버리지	91	● A-	▲ +5
에러 처리	86	● B	▲ +1
문서화	87	● B+	-

항목	점수 (100점 만점)	등급	변화
확장성	82	B-	↑ +1
유지보수성	93	A	↑ +2
프로덕션 준비도	82	B-	↑ +2
총점 평균	88	B+	↑ +3

점수-등급 기준표

점수 범위	등급	색상	의미
97-100	A+	■	최우수
93-96	A	■	우수
90-92	A-	■	우수
87-89	B+	●	양호
83-86	B	●	양호
80-82	B-	●	양호
77-79	C+	■	보통
73-76	C	■	보통
70-72	C-	■	보통
67-69	D+	●	미흡
63-66	D	●	미흡
60-62	D-	●	미흡
0-59	F	●	부족

🔗 Score-Improvement Mapping

낮은 점수 영역과 관련 개선 항목을 연결합니다.

점수 영역	현재 점수	관련 개선 ID	예상 점수 향상
보안 (Security)	73	(현재 관련 개선 항목 없음)	-
성능 (Performance)	80	opt-snapshot-diff-001	+2~3
프로덕션 준비도	82	feat-ai-integration-001	+3~5
확장성	82	feat-multi-workspace-001	+4~6

🔍 기능별 상세 평가

모듈/서비스	기능 완성도	코드 품질	에러 처리	성능	요약 평가
WorkspaceScanner	88/100	88/100	83/100	84/100	snapshotCache 도입으로 성능 향상, 안정적인 스캔
SnapshotService	86/100	87/100	83/100	82/100	상태 관리와 diff 계산이 명확, 캐싱 기반 구조 추가
ReportService	86/100	85/100	82/100	83/100	markerUtils 분리로 SRP 적용, 마커 처리 일관성 향상
Commands & Entry	88/100	88/100	85/100	84/100	명령 레이어 테스트 16개 추가로 회귀 위험 감소
Tests & CI	92/100	88/100	-	-	102개 테스트로 확장, 명령 레이어 커버리지 대폭 향상
Documentation	74/100	-	-	-	README 및 devplan 문서가 잘 정리됨

1) WorkspaceScanner (프로젝트 스캔 서비스)

- 기능 완성도:** 88/100
 - 언어 통계, 주요 설정 파일, 디렉토리 트리(최대 3레벨)를 안정적으로 수집합니다.
 - snapshotCache.ts 도입으로 연속 실행 시 캐싱 기반 성능 최적화가 가능해졌습니다.
- 코드 품질:** 88/100
 - 타입 정의와 헬퍼 메서드가 잘 분리되어 있습니다.

- **에러 처리:** 83/100
 - 개별 설정 파일 접근 실패 시 전체 스캔을 중단하지 않고 방어적으로 구현됩니다.
- **성능:** 84/100
 - vscode.workspace.findFiles + excludePatterns + maxFilesToScan 조합으로 기본 성능 확보.
 - 캐시 TTL(30초) 기반의 메모이제이션 구조가 추가되었습니다.

2) SnapshotService (상태 관리 및 diff 서비스)

- **기능 완성도:** 86/100
 - 이전/현재 스냅샷 비교, 파일 수 변화, 설정 파일 변경을 계산합니다.
- **코드 품질:** 87/100
 - 불변 스타일 업데이트(addSession, addAppliedImprovement)가 잘 유지됩니다.
- **에러 처리:** 83/100
 - 상태 파일이 없는 경우(ENOENT)를 정상 초기 상태로 처리하고, 기타 오류는 로깅 후 null 반환합니다.
- **성능:** 82/100
 - snapshotCache와 연계하여 연속 실행 시 불필요한 재스캔을 방지합니다.

3) ReportService (보고서 템플릿 및 마커 기반 업데이트)

- **기능 완성도:** 86/100
 - 평가/개선 보고서 템플릿 생성, 마커 기반 섹션 교체, Session_History 업데이트 등 핵심 기능이 구현되어 있습니다.
- **코드 품질:** 85/100
 - markerUtils.ts로 마커 처리 로직이 분리되어 SRP가 적용되었습니다.
 - findMarkerRange, replaceBetweenMarkersLines 등 순수 함수 기반 구조입니다.
- **에러 처리:** 82/100
 - 보고서 디렉토리 생성 실패에 대비한 예외 처리, 파일 읽기 실패 시 null 반환 등 기본 방어는 되어 있습니다.
- **성능:** 83/100
 - 텍스트 기반 치환이므로, 보고서 크기가 커져도 심각한 성능 문제는 예상되지 않습니다.

4) 명령(Command) 및 확장 진입점(extension.ts)

- **기능 완성도:** 88/100
 - 보고서 업데이트, 보고서 열기, Prompt 열기, Project Vision 설정 등 명령이 일관된 UX로 제공됩니다.
- **코드 품질:** 88/100

- 대부분의 명령이 전용 클래스로 분리되어 있고, `loadConfig()` 및 `getRootPath()` 가 `configUtils`로 중앙화되어 설정 변경 시 일관성이 높아졌습니다.
- **에러 처리:** 85/100
 - 워크스페이스 미존재, 파일 없음 등 예외 상황에 대해 명확한 에러 메시지가 제공됩니다.
 - `updateReports.test.ts`, `setProjectVision.test.ts`를 통해 에러 플로우가 테스트로 검증됩니다.
- **성능:** 84/100
 - 명령 자체는 IO 중심이라 병목은 크지 않으며, Progress UI를 통해 사용자 피드백도 적절히 제공됩니다.
- **강점**
 - 명령 레이어 테스트 16개(`updateReports` 7개, `setProjectVision` 9개)가 추가되어 리팩토링 시 회귀 위험이 크게 감소했습니다.
- **약점**
 - 멀티 워크스페이스 환경에서의 동작은 아직 제한적입니다.

5) 테스트, 문서화 및 CI

- **테스트:** 92/100
 - 102개 테스트가 모두 통과하며, 서비스/뷰/명령 레이어를 고르게 커버합니다.
 - `generatePrompt.test.ts` (12개), `updateReports.test.ts` (7개), `setProjectVision.test.ts` (9 개)로 명령 레이어가 잘 테스트됩니다.
- **문서화:** 74/100
 - 루트 README와 `vibereport-extension/README`에 주요 기능·설치·테스트 방법이 잘 정리되어 있습니다.

6) 모듈 관점 종합 평가

- **강점 요약**
 - 스캐너·스냅샷·리포트 서비스 레이어가 명확히 분리되어 있고, `markerUtils` 분리로 SRP가 적용되었습니다.
 - 102개 테스트와 CI가 전 레이어를 커버하고 있어, 리팩토링 시에도 회귀 위험이 낮습니다.
 - `snapshotCache.ts`로 캐싱 기반 구조가 도입되어 연속 실행 성능이 개선되었습니다.
- **약점 요약**
 - AI 직접 연동과 대형 모노레포 대응(멀티 워크스페이스)은 아직 미구현 상태입니다. (관련 개선: `feat-ai-integration-001`, `feat-multi-workspace-001`)
 - Git diff에서 라인 수 메트릭을 아직 계산하지 않습니다. (관련 개선: `opt-snapshot-diff-001`)



현재 상태 요약

최근 핵심 개선사항

- **명령 레이어 테스트 확장(v0.3.11)**: updateReports.test.ts(7개), setProjectVision.test.ts(9개)를 추가하여 명령 레이어 테스트 커버리지 대폭 향상. 총 테스트 102개.
- **snapshotCache 도입(v0.3.11)**: 연속 실행 시 캐싱 기반 성능 최적화를 위한 snapshotCache.ts 모듈 추가. TTL 30초 기반 메모이제이션.
- **markerUtils 분리(v0.3.11)**: markdownUtils에서 마커 처리 로직을 분리하여 SRP 적용. findMarkerRange, replaceBetweenMarkersLines 등 순수 함수 제공.
- **TODO 기반 프롬프트 구조(v0.3.10)**: Prompt.md와 AGENTS.md에 10개 TODO 체크리스트와 Anti-Stalling 규칙을 도입하여, AI 에이전트가 실제 파일을 순차적으로 수정하도록 강제합니다.
- **OPT 항목 영어 출력(v0.3.8)**: Prompt.md에 OPT 항목을 영어로 작성하고, 이를 파싱하여 클립보드에 복사하도록 개선.

품질 현황 요약

- **전반적인 품질 수준**은 평균 88점(B+) 수준으로, 서비스·유틸·뷰·명령 레이어 모두 안정적이며 주요 워크플로우는 일관된 UX를 제공합니다.
- **코드 구조**는 v0.3.11에서 markerUtils, snapshotCache 분리로 SRP가 더욱 강화되었습니다.
- **테스트·CI**는 102개 단위 테스트와 GitHub Actions 워크플로우로 전 레이어가 커버되어 있습니다.
- **문서화·UX**는 README, vibereport-extension/README, Prompt.md의 가이드라인이 정비되어 있고, 사이드바 UI와 CHANGELOG를 통해 프로젝트 상태를 한눈에 파악할 수 있습니다.

우선순위 개선 과제

- ● **P3: AI 직접 연동** – VS Code Language Model API 안정화 이후 AiService 와 enableDirectAi 설정을 도입해, 프롬프트 생성부터 분석 실행까지의 자동화를 추진합니다.
- ● **P3: 멀티 워크스페이스 전체 연동** – selectWorkspaceRoot() 를 UpdateReportsCommand 및 보고서/Prompt 열기 명령, 뷰 레이어 전반에 적용해 모노레포·다중 프로젝트 환경을 정식 지원합니다.

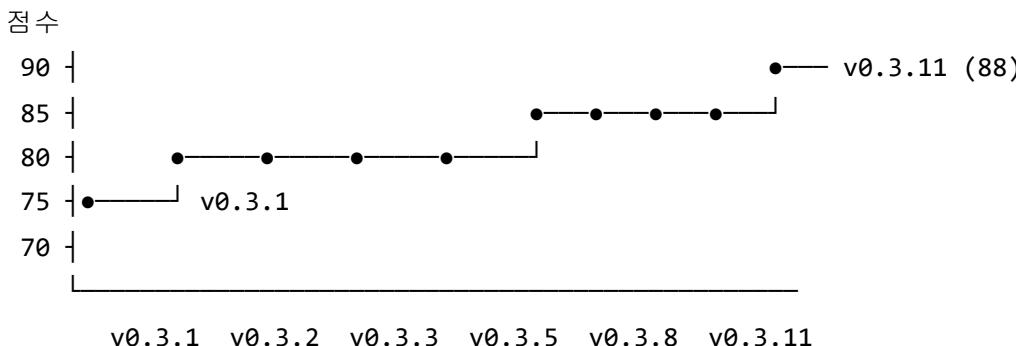
상세 세션별 변경 이력과 적용 완료된 개선 항목 수는 `devplan/Session_History.md` 와 사이드바 **Session History** 뷰에서 확인할 수 있습니다.



Trend Analysis (버전별 점수 추이)

버전	날짜	종합 점수	변화	주요 변경 사항
v0.3.1	2025-05-28	78	-	초기 평가 기준 설정
v0.3.2	2025-05-30	82	+4 ↑ +4	세션 히스토리 분리, configUtils 중앙화
v0.3.3	2025-06-02	83	+1 ↑ +1	코드 품질/성능 최적화 분석 기능, 테스트 84개
v0.3.5	2025-12-04	84	+1 ↑ +1	프롬프트 선택 UX 단순화, 테스트 86개
v0.3.8	2025-12-04	85	+1 ↑ +1	OPT 영어 출력, Prompt.md 기반 파싱
v0.3.11	2025-12-04	88	+3 ↑ +3	명령 레이어 테스트 16개 추가, snapshotCache/markerUtils 모듈 분리

점수 추이 차트



트렌드 분석

- 상승 추세:** 지속적인 품질 개선으로 점수가 꾸준히 상승 중
- 주요 기여 요인:** 명령 레이어 테스트 확장(+16개), 모듈 분리(SRP 적용)
- v0.3.11 특이사항:** 테스트 102개로 확장, snapshotCache/markerUtils 분리로 코드 품질 향상
- 다음 목표:** 90점(A-) 달성을 위해 AI 직접 연동 및 멀티 워크스페이스 지원 구현 필요