

Product Plan 템플릿 사용 가이드

개요

시작하기 전에: PO(혹은 기획자)의 마인드셋

문제 정의

고객 요구사항 정의(CRS) 작성

요구사항 구체화

사용자 흐름 확인 (User Journey/Flow)

정보 구조 설계 (IA)

와이어프레임 작업

화면 설계서 작업

개발 요구사항 명세 작성 (SRS)

데이터 구조 및 API 후보 정의

마치며

부록 - 템플릿 샘플

유즈케이스

유저 저니

서비스 기획 여정 가이드

개요

이 문서는 [\[TPL\] Product Plan 템플릿](#)을 처음 사용하는 분들을 위한 가이드입니다. 기획은 정해진 답이 없는 과정이지만, 이 가이드를 통해 전체 프로세스를 이해하고 템플릿을 채워나가는 데 도움을 받으시길 바랍니다.

시작하기 전에: PO(혹은 기획자)의 마인드셋

- 기획 문서는 살아있어야 합니다

◦ 기획은 한번에 끝나는 폭포수 모델이 아닙니다(폭포수 방법론이 사용되고 있는 경우에도 그렇습니다). 화면설계서가 나온 상태에서도 페르소나를 다시 수정할 수 있고, 후속 단계에서 새로운 아이디어가 나오면 다시 문서를 업데이트해야 합니다. 기획 문서는 항상 살아있는 문서여야 하며, 계속 발전시켜 나가는 것이 기획자의 역할입니다.

- 상황에 따라 유연하게 대처하세요

◦ 이 가이드는 정석적인 프로세스일 뿐, 항상 정답은 아닙니다. 시장 상황이 급변하거나, 최소한의 기능으로 빠르게 시장 반응을 봐야 할 때(MVP)는 일부 단계를 과감히 생략하거나 통합할 수 있습니다. '완벽한 기획'을 위한 계획이 아니라, '성공적인 제품/서비스'를 위한 계획을 세우는 것임을 잊지 마세요. 교조주의적으로 방법론/템플릿을 따라갈 필요는 없습니다.

- 기획은 '소통'입니다

◦ 최고의 기획서는 멋진 문서가 아니라, 개발자와 디자이너가 완벽하게 이해하고 공감한 기획서입니다. 문서를 전달하는 것으로 역할이 끝났다고 생각하지 마세요. 수시로 질문하고, 진행 상황을 공유하며, 함께 문제를 해결해 나가는 '소통의 중심'에 기획자가 있습니다.

▣ 상세 단계의 작업법에 대해 의문사항이 있거나, 구체적인 템플릿을 필요로 하는 경우, 동료에게 요청해 주세요. 도움 요청을 받은 사람은 기존 산출물 등을 통해 예시를 충실히 제공해 주세요.

문제 정의

➊ 무엇을 만들 것인가?

모든 기획은 '이걸 왜 만들어야 하지?'라는 질문에서 시작합니다. 이 단계에서는 아직 템플릿을 채우기보다, 아이디어를 구체화하고 우리가 해결하려는 문제가 무엇인지 명확히 하는 데 집중합니다.

고객 요구사항 정의(CRS) 작성

- ❶ 고객이 무엇을 원하는지, 어떤 문제를 겪고 있는지 파악하고 문서화하는 단계입니다:

분류	설명										
CRS란?	<p>고객의 요구사항을 구체적인 명세(CRS, Customer Requirement Specification)로 정리합니다. 만약 고객의 요구가 불명확하다면, 가상의 고객(User Persona)을 설정하여 그들이 겪을 법한 문제를 추정하고 필요사항을 정의할 수 있습니다.</p> <p>가상의 고객은 다음과 같이 구체적으로 정의할 수도 있으나, 임의로 필요한 수준으로만 정의하여도 무방합니다.</p> <table border="1"><thead><tr><th>구분</th><th>내용</th></tr></thead><tbody><tr><td>이름</td><td>홍길동</td></tr><tr><td>개요</td><td>나이, 직업, 거주지, 학력, 가족 관계 등</td></tr><tr><td>한줄 묘사</td><td>페르소나의 핵심 욕망이나 성격을 한 마디로 표현</td></tr><tr><td>구체 목표</td><td>우리 제품/서비스를 통해 이루고 싶어하는 구체적인 목표</td></tr></tbody></table>	구분	내용	이름	홍길동	개요	나이, 직업, 거주지, 학력, 가족 관계 등	한줄 묘사	페르소나의 핵심 욕망이나 성격을 한 마디로 표현	구체 목표	우리 제품/서비스를 통해 이루고 싶어하는 구체적인 목표
구분	내용										
이름	홍길동										
개요	나이, 직업, 거주지, 학력, 가족 관계 등										
한줄 묘사	페르소나의 핵심 욕망이나 성격을 한 마디로 표현										
구체 목표	우리 제품/서비스를 통해 이루고 싶어하는 구체적인 목표										
왜 작성해야 하는가?	우리가 만들 서비스가 실제로 사용될지, 고객의 문제를 해결해 줄 수 있을지 검증하는 가장 중요한 첫걸음입니다.										
템플릿의 어디에 있는가?	<p>이 단계의 산출물은 고객이나 전사 기획 등으로부터 주어지는 것을 원칙으로 합니다(다만, 명시적으로 당 조직에 CRS 작성의 권한과 역할이 부여되어 있거나, 여타 이유로 CRS 작성과 관리가 필요하다고 판단되는 경우에는 직접 작성합니다).</p> <p>따라서 템플릿에 직접적으로 포함되지는 않지만, 이후 모든 단계의 근거 자료가 됩니다.</p> <p>▣ CRS를 관리할 필요가 있다고 판단되는 경우, [T PL] Product Plan 템플릿 템플릿 이용시 하위 페이지로 'CRS' 페이지를 추가해 주세요.</p>										
작성 예시	<ul style="list-style-type: none">PV25 요구사항 정의HC24 CRS										

요구사항 구체화

- ⓘ 추상적인 요구사항을 실제 사용 시나리오로 구체화하는 단계입니다.

분류	설명
어떤 단계인가?	<p>유즈케이스, 도메인 스토리텔링, 이벤트 스토밍 등과 같은 기법을 활용하여 '누가, 언제, 어디서, 무엇을, 어떻게' 하는지에 대한 이야기를 만듭니다.</p> <p>기법(도구)가 중요한 것이 아닙니다. 구체화를 해내는 것 자체에 의의를 두세요.</p>
왜 작성해야 하는가?	개발팀이 사용자의 입장에서 서비스의 동작 방식을 명확하게 이해하고, 놓치는 부분을 줄일 수 있습니다.
템플릿의 어디에 있는가?	<p>이 단계에서 도출된 시나리오들은 템플릿의 '주요 프로세스 일람'에 다이어그램(플로우차트, 시퀀스 다이어그램 등) 형태로 표현될 수 있습니다.</p> <p>▣ 정책이란 무엇인가 도 참고하세요.</p>
작성 예시	<ul style="list-style-type: none">유즈케이스: ▣ Product Plan 템플릿 사용 가이드 유즈케이스도메인 스토리텔링: ▣ ETSI 전체 도메인 설계이벤트 스토밍: ▣ 이벤트 스토밍

사용자 흐름 확인 (User Journey/Flow)

- ⓘ 사용자가 서비스를 이용하는 전체적인 여정이나 특정 기능의 흐름을 시각적으로 확인합니다.

분류	설명
어떤 단계인가?	사용자가 서비스에 진입해서 목표를 달성하고 이탈하기 까지의 과정을 순서대로 그립니다.
왜 작성해야 하는가?	서비스의 흐름이 자연스러운지, 사용자가 불편을 겪을 만한 지점은 없는지 미리 검토할 수 있습니다.
템플릿의 어디에 있는가?	요구사항 구체화 단계와 마찬가지로, '주요 프로세스 일람' 섹션에 '유저 저니'나 '플로우차트' 형태로 작성합니다.
작성 예시	<ul style="list-style-type: none">▣[타피스] 앱 플로우▣유저 저니

정보 구조 설계 (IA)

- ⓘ 서비스가 제공할 정보의 전체 구조와 메뉴 체계를 설계합니다.

분류	설명

어떤 단계인가?	웹사이트의 '사이트맵'처럼, 어떤 메뉴가 있고 각 메뉴는 어떤 하위 메뉴로 구성되는지 계층 구조로 표현합니다.
왜 작성해야 하는가?	사용자가 원하는 정보를 쉽고 빠르게 찾을 수 있도록 빠대를 잡는 과정입니다.
템플릿의 어디에 있는가?	'정보 아키텍처' 섹션에 Mermaid나 다른 도구를 사용하여 IA를 작성합니다.
작성 예시	<ul style="list-style-type: none"> • EU20 정보 아키텍처 • PV25 - UI 기획

와이어프레임 작업

❶ 화면의 대략적인 레이아웃과 구성 요소를 스케치합니다.

⚠️ 라이브러리성 제품 등, 화면 구성 요소가 없는 제품/서비스의 경우 생략될 수 있습니다.

분류	설명
어떤 단계인가?	디자인 요소를 최대한 제외하고, '어떤 버튼이 어디에 있는지', '어떤 정보가 표시되는지'에 집중하여 간단한 화면 구조를 그립니다.
왜 작성해야 하는가?	본격적인 디자인 작업에 들어가기 전, 화면의 기능과 구조에 대한 빠른 합의를 이끌어낼 수 있습니다. 디자인이 중요하지 않은 제품/서비스의 경우, 와이어프레임만 제공하고, 디자인은 개발팀이 직접 결정하여도 무방합니다.
템플릿의 어디에 있는가?	'화면 기획' 섹션에 와이어프레임 파일을 첨부하거나 링크를 기재합니다. 이 때 생성한 화면들의 목록은 '화면 목록' 섹션과 일치해야 합니다. 파일을 첨부할 경우, 버전 관리 테이블을 추가합니다.
작성 예시	<ul style="list-style-type: none"> • PV25 - UI 기획

⚠️ 아래 작업은 기획 담당자가 필수로 수행해야 하는 역할은 아닙니다.

기획자가 내용을 채우기 어려운 경우, 디자인 조직과의 협업, 혹은 디자인 조직에의 인계를 통해 완성합니다.

화면 설계서 작업

❷ Lofi 와이어프레임만으로 설명이 부족할 경우, 더 상세한 화면 설계서를 작성합니다.

분류	설명

어떤 단계인가?	각 화면의 모든 케이스(정상, 예외, 오류 등)에 대한 정책, 인터랙션, 상세한 기능 명세를 담은 문서를 작성합니다.
왜 작성해야 하는가?	복잡한 기능이나 정책이 포함된 화면에 대해 개발, 디자인, QA팀이 모두 동일한 내용을 이해하고 작업할 수 있도록 합니다.
템플릿의 어디에 있는가?	'화면 기획' 및 '검증' 섹션을 활용하여 상세한 시나리오와 정책을 문서화할 수 있습니다.
작성 예시	-

⚠ 아래 작업은 기획 담당자가 필수로 수행해야 하는 역할은 아닙니다.

기획자가 내용을 채우기 어려운 경우, 개발 조직과의 협업, 혹은 개발 조직에의 인계를 통해 완성합니다.

개발 요구사항 명세 작성 (SRS)

➊ 개발팀이 보고 실제 개발을 진행할 수 있는 기술적인 명세를 작성합니다.

분류	설명
어떤 단계인가?	기능 요구사항(FR)과 비기능 요구사항(NFR)을 포함하는 SRS를 작성합니다.
왜 작성해야 하는가?	기획자와 개발자 간의 오해를 줄이고, 개발의 목표와 범위를 명확하게 설정합니다.
템플릿의 어디에 있는가?	SRS는 개발 조직이 기획자료를 인계받아 작성하는 것을 기본으로 합니다(협의에 따라 변경 가능). 따라서 개발 팀 담당의 [TPL] 제품 및 프로젝트 템플릿 하위에 템플릿이 있습니다.
작성 예시	• [PV25 요구사항 정의]

데이터 구조 및 API 후보 정의

➋ 화면에 필요한 데이터는 무엇이고, 서버와 어떻게 통신해야 할지 정의합니다.

분류	설명
어떤 단계인가?	각 화면에 표시될 데이터 항목(예: 사용자 이름, 상품 가격)을 정의하고, 이를 바탕으로 필요한 API의 후보 목록(예: 사용자 정보 조회 API, 상품 목록 조회 API)을 작성합니다. 제품/서비스가 라이브러리성이거나, 백엔드 Rest API 중심인 경우, 다른 단계보다 이 단계의 중요성이 높습니다. 기획에서 기술적인 결정을 하기 어렵더라도, 개발 조직이 합리적인 의사결정을 할 수 있도록 충분한 백데이터

	를 제공해 주어야 합니다. 이 경우, 고객 요구사항 정의(CRS) 작성 , 요구사항 구체화 , 사용자 흐름 확인 (User Journey/Flow) 를 충실히 작성하여야 합니다.
왜 작성해야 하는가?	개발자가 헤더 파일/OpenAPI 등 인터페이스, 데이터베이스(ERD) 설계를 원활하게 진행할 수 있도록 돋는 중요한 입력 자료가 됩니다.
템플릿의 어디에 있는가?	이 단계의 산출물은 템플릿에 직접적으로 포함되지는 않습니다. 고객 요구사항 정의(CRS) 작성 , 요구사항 구체화 , 사용자 흐름 확인 (User Journey/Flow) 에 충실히 자료를 기입합니다.
작성 예시	• PV25 개발 상세 설계 및 구현

마치며

이 가이드가 여러분의 성공적인 기획 여정에 작은 보탬이 되기를 바랍니다. 처음에는 막막하게 느껴질 수 있지만, 각 단계를 차근차근 밟아가다 보면 어느새 완성된 기획서를 마주하게 될 것입니다.

부록 - 템플릿 샘플

유즈케이스

▼ Click here to expand...	
구분	내용
Identifier	SAMS_UC1
Objective	협력업체가 ECU SW 개발 완료 후, 보유한 인증서·개인키로 SW에 서명하여 TDM(SAMS)에 제출하고 검증을 거쳐 결과를 반환받음
Description	<ul style="list-style-type: none"> 협력업체는 이미 발급받은 서명 인증서(ECDSA 2048bit)와 개인키를 사용해 ECU SW를 서명한다 CSMS Portal을 통해 SW와 서명을 SAMS에 업로드하면, SAMS는 인증서 체인 유효성, 자격 검증, 서명값 검증을 수행한다 검증 성공 시 OK를 반환하며, 실패 시 오류 코드와 함께 FAIL을 반환한다
Configuration	<ul style="list-style-type: none"> Portal ↔ SAMS 간 통신은 TLS 1.3 + mTLS 채널 사용 협력업체 서명 인증서는 TDM Root CA → Sub CA 체인으로 검증 가능해야 함

	<ul style="list-style-type: none"> 협력업체 서명 개인키는 로컬 환경에 AES-256-GCM으로 암호화되어 보관, HSM 내부 마스터 키로 키 래핑
Conditions	<ul style="list-style-type: none"> 협력업체 계정 및 권한 사전 등록 협력업체 보유 인증서가 유효하며 폐기되지 않음 ECU SW 바이너리와 서명값 준비 완료
REQ	<p>⚠ UC와 SRS의 추적성 관리용(SRS가 없으므로 예시임)</p> <ul style="list-style-type: none"> UC 기반으로 SRS가 작성된 뒤 기재하면 됨 혹은 생략해도 무방함 <ul style="list-style-type: none"> SRS_SW_SIGN_1 (SW 서명 검증) SRS_PKI_CHAIN_1 (인증서 체인 검증) SRS_PKI_CRL_1 (폐기 상태 검증)
Related API	<p>⚠ API는 개발팀이 작성해도 무방하나, 개발팀이 API작성에 충분한 정보를 가질 수 있도록 나머지 정보는 다 채워주어야 함.</p> <ul style="list-style-type: none"> PV25 TDM - package verify
Sequence Diagram	<pre> sequenceDiagram participant Supplier as 협력업체 participant SAMS as SAMS Supplier->>SAMS: SW와 서명 제출 (SW, 서명) SAMS-->>Supplier: (SW, 서명) 검증 Note over Supplier,SAMS: ECU SW 서명 및 제출 시나리오 (응답시간 5초 이내) alt [검증 성공] SAMS-->>Supplier: OK (성공 코드) else [검증 실패] SAMS-->>Supplier: FAIL (오류 코드) Supplier->>SAMS: 제출 종료 SAMS-->>Supplier: 문제 수정 후 재시도 end </pre>
<p>▼ Click here to expand...</p> <pre> 1 sequenceDiagram 2 participant Supplier as 협력업체 3 participant SAMS as SAMS 4 5 Note over Supplier,SAMS: ECU SW 서명 및 제출 시나리오 (응답시간 5 초 이내) 6 </pre>	

```
7 Supplier->>SAMS: SW와 서명 제출 (SW, 서명)
8 SAMS->>SAMS: (SW, 서명) 검증
9 alt 검증 성공
10     SAMS->>Supplier: OK (성공 코드)
11     Supplier->>Supplier: 제출 종료
12 else 검증 실패
13     SAMS->>Supplier: FAIL (오류 코드)
14     Supplier->>Supplier: 문제 수정 후 재시도
15 end
```

Step	Actor	Description	Result
1	Supplier	협력업체가 SW를 서명하여 (SW, 서명) 쌍을 SAMS에 제출한다	
2	SAMS	SAMS는 (SW, 서명) 쌍을 검증한다	성공/실패
3	SAMS	SAMS는 응답코드를 협력업체에 내려준다	
4-1	Supplier	성공시: 협력업체는 제출을 종료한다	
4-2	Supplier	실패시: 협력업체는 문제사항 수정 후 프로토콜을 재시도한다	
Remark	응답시간 5초 이내 (평균 3초 목표)		

유저 저니

```
1 journey
2   title 기획자 J의 업무 여정
3
4   section 시작하기
5     문제 정의: 5: 기획자 J
6     고객 이해 (CRS): 4: 기획자 J
7
8   section 개요 잡기
9     정보 구조 설계 (IA): 4: 기획자 J
10    화면 목록 작성: 3: 기획자 J
11    요구사항 구체화: 4: 기획자 J
12    사용자 흐름 확인: 4: 기획자 J, 디자이너 D
13
14   section 규칙과 질서
15     정책 수립: 3: 기획자 J
16     API 후보 정의: 3: 기획자 J, 개발자 K
17
18   section UI/UX
19     와이어프레임 스케치: 3: 기획자 J, 디자이너 D
20
21   section 개발팀에게로
22     기술 명세 작성 (SRS): 3: 기획자 J, 개발자 K
```