
REMOTE WORK GUIDEBOOK

인도 개발자 협업 가이드북

CONTENTS

INTRO	인도 인재 유치의 필요성	04
PART 1	인도 인재 유치의 주요 이유	06
1.	우수한 인재 풀	07
2.	뛰어난 기술력과 실무 역량	07
3.	경제적 효율성	08
4.	글로벌 커뮤니케이션 능력	08
5.	원격 협업 경험	09
6.	글로벌 리더십 입증	09
7.	풍부한 인력 공급	09
PART 2	인도 인력 채용 방법	10
1.	인도 개발자를 '인도 현지'에서 원격으로 고용하는 방법	11
1.1	직접 고용	11
1.1.1	근로계약	12
1.1.2	프리랜서 계약	12
1.1.3	근로계약과 프리랜서 계약의 주요 차이	13
1.1.4	실제 직접 채용 시 유의 사항	14
1.2.	인도 현지 법인 설립	15
1.3.	기록상 고용주 서비스 활용	16
1.4.	인도 아웃소싱 업체 활용	17
2.	인도 개발자를 '한국으로 초청'하여 국내에서 함께 근무하는 방법	18
2.1	E-7 비자를 통해 정규직 또는 계약직으로 고용	18
2.2	E-7 비자 발급 관련 가이드 라인	19
2.2.1	E-7 비자 개요	19
2.2.2	E-7 비자 발급 필요 요건	20
2.2.3	주요 절차	21
2.2.4	비자 발급 서류 및 요건	21
2.2.5	상세 절차 및 소요 시간	22

PART 3	인도 개발자와 원격으로 일하는 방법	25
1.	원격 근무 도전 과제	27
2.	원격 근무 가이드 베스트 프랙티스	28
2.1	안정적인 전기·통신 인프라 준비	28
2.2	개발자에게 하드웨어 제공	29
2.3	원격 근무 시간 확립	32
2.4	개발자와 커뮤니케이션	34
2.5	영어 커뮤니케이션	39
2.6	인도 개발자의 커뮤니케이션 방식 차이	42
2.7	보안 관점에서 고려해야 할 점	46
PART 4	글로벌 애자일 팀 운영	48
1.	애자일 방법론	50
1.1	스크럼의 3가지 역할	51
1.2	스크럼의 5가지 이벤트	52
1.3	스크럼의 3가지 산출물	53
2.	코드 리뷰 및 버전 관리	54
2.1	코드 리뷰	54
2.2	버전 관리	56
PART 5	성공적인 원격 근무를 위한 모범 사례	58
PART 6	인도 개발자 온보딩 체크리스트	64
1.	채용 전 체크리스트	65
2.	채용 후 첫 주 체크리스트	66
3.	첫 달 체크리스트	66
OUTRO	마무리 글	68

인도 인재 유치의 필요성

디지털 시대의 가속화는 한국 기업들, 특히 중소기업들이 글로벌 경쟁력을 확보하기 위해 우수한 IT 인재의 확보를 절실히 요구하고 있습니다. 그러나 국내에서 글로벌 스탠다드에 부합하는 인재를 찾는 데에는 한계가 존재합니다. 이에 따라 많은 기업들이 인도, 베트남 등 해외의 우수 개발자 채용을 적극적으로 고려하고 있습니다. 하지만 글로벌 개발자와의 협업 과정에서 계약 방식부터 채용 후 협업 방법에 이르기까지 명확한 지침이 부족해 어려움을 겪는 경우가 많습니다. 심지어 어디서부터 시작해야 할지 모르는 기업들도 존재합니다.

본 책자는 이러한 문제를 해결하고자, 인도 개발자 채용의 주요 이유부터 다양한 채용 방법, 효과적인 커뮤니케이션 전략, 보안 관점에서의 고려사항, 글로벌 애자일 팀 운영 방법, 성공적인 원격 근무를 위한 모범 사례, 그리고 인도 개발자 온보딩 체크리스트까지 전반적인 내용을 다루고 있습니다. 특히, 원격 근무에 익숙하지 않은 기업들을 위한 성공적인 원격 근무 가이드라인을 제공함으로써, 인도 개발자와의 협업을 원활하게 진행할 수 있도록 돕고자 합니다.

글로벌 인재와의 원격 근무는 현대 비즈니스 환경에서 점점 더 중요해지고 있으며, 이를 성공적으로 수행하기 위해서는 체계적인 준비와 관리가 필수적입니다. 인도 개발자의 풍부한 인재 풀과 뛰어난 기술력을 활용함으로써 기업은 경제적 효율성과 글로벌 커뮤니케이션 능력을 동시에 강화할 수 있습니다. 이를 위해서는 적절한 채용 방법과 온보딩 절차를 통해 인도 개발자와의 협업을 원활하게 진행해야 하며, 효과적인 커뮤니케이션 전략과 보안 가이드라인을 준수하여 안전하고 효율적인 작업 환경을 조성해야 합니다.

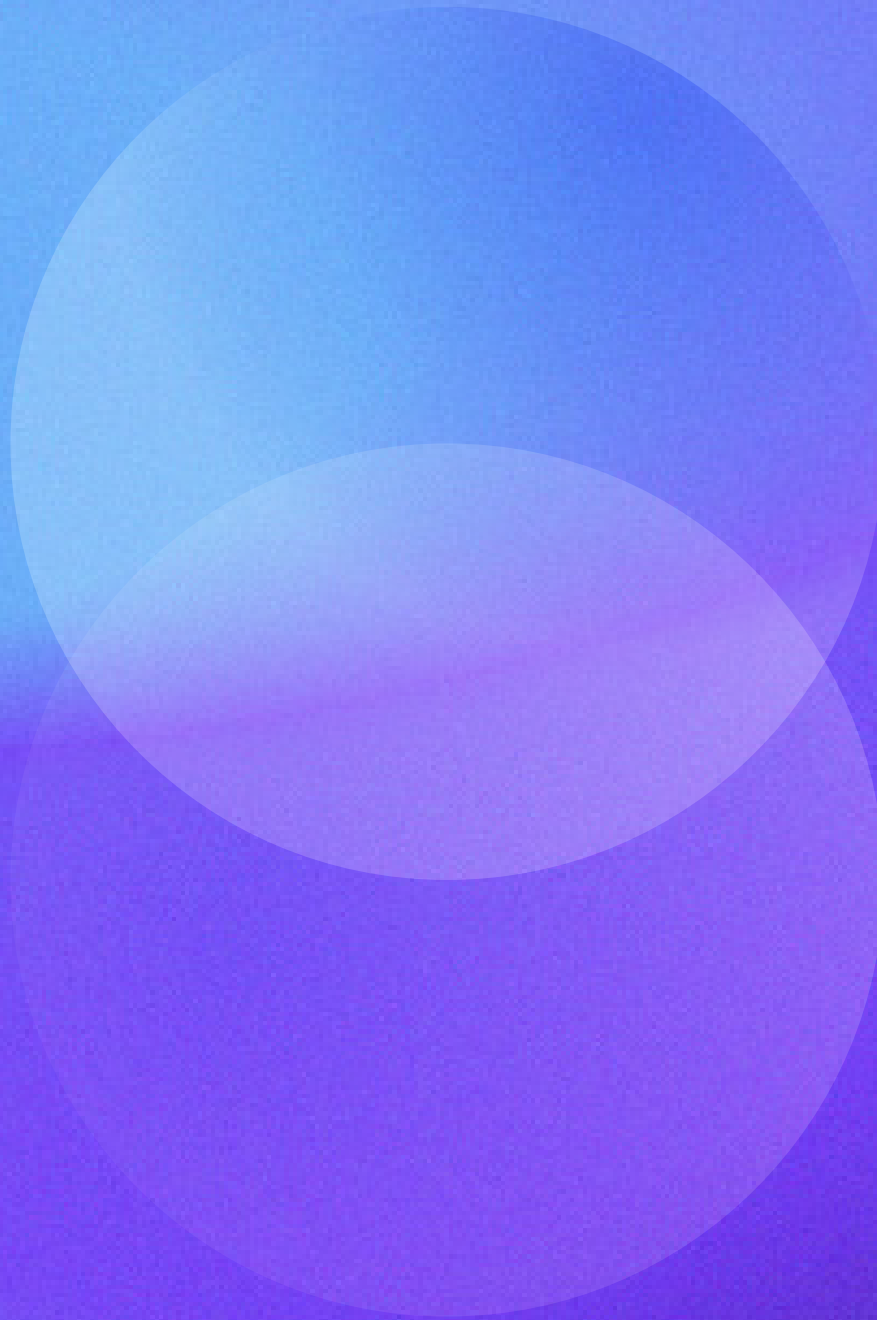
또한, 애자일 방법론을 기반으로 한 글로벌 팀 운영과 코드 리뷰 및 버전 관리의 체계적인 적용은 프로젝트의 성공과 팀의 생산성을 높이는 데 중요한 역할을 합니다. 성공적인 원격 근무를 위해서는 안정적인 인프라 구축과 팀 빌딩 활동을 통해 팀원 간의 유대감을 강화하고, 지속적인 보안 관리와 인적 보안 교육을 통해 보안 위협을 최소화해야 합니다.

이 가이드북이 한국 기업들이 인도 개발자와의 협업을 성공적으로 이끌고, 글로벌 경쟁력을 강화하는 데 실질적인 도움이 되기를 바랍니다. 원격 근무 환경에서 높은 생산성과 협업 효율성을 유지하며, 인도 개발자와의 협업을 통해 조직의 목표를 효과적으로 달성하고, 글로벌 시장에서의 입지를 더욱 공고히 할 수 있기를 기대합니다. 글로벌 인재와의 협업은 단순한 인력 구성의 변화가 아니라, 더 나은 업무 환경과 더 큰 성과를 이끌어내는 혁신의 기회입니다. 이를 통해 조직과 구성원 모두가 성장하고 발전할 수 있기를 바랍니다.

앞으로도 변화하는 글로벌 시장 속에서 유연하고 효과적인 협업 방식을 지속적으로 모색하며, 다양한 문화와 배경을 가진 인재들과 함께 성공적인 프로젝트를 완수해 나가시길 응원합니다. 본 가이드북이 여러분의 여정에 든든한 동반자가 되기를 바랍니다.

PART 1

인도 인재 유치의 주요 이유



01

인도 인재 유치의 주요 이유

- ☐ 우수한 인재 풀
- ☐ 뛰어난 기술력과 실무 역량
- ☐ 경제적 효율성
- ☐ 글로벌 커뮤니케이션 능력
- ☐ 원격 협업 경험
- ☐ 글로벌 리더십 입증
- ☐ 풍부한 인력 공급

01

우수한 인재 풀

인도는 세계에서 두 번째로 많은 인구를 보유한 국가로, 청년층이 다수인 14억 인구 중 상당수가 IT 및 소프트웨어 산업에 종사하고 있습니다. 인도 정부는 STEM(과학, 기술, 공학, 수학) 교육을 중심으로 지원하여 지속적으로 세계적 수준의 소프트웨어 인재를 양성하고 있으며, 이러한 인재 풀은 한국 기업들이 요구하는 기술적 역량을 충족시킬 수 있습니다.

02

뛰어난 기술력과 실무 역량

인도의 소프트웨어 엔지니어들은 높은 기술적 능력과 문제 해결 역량을 갖추고 있습니다. IIT(Indian Institutes of Technology)와 NIT(National Institutes of Technology)와 같은 인도 최고의 교육 기관들은 매년 다수의 글로벌 수준의 소프트웨어 엔지니어를 배출하고 있으며, 이들은 글로벌 IT 프로젝트와 다양한 실무 경험을 통해 뛰어난 성과를 창출할 수 있는 역량을 갖추고 있습니다.

경제적 효율성

인도 개발자를 채용하는 것은 국내 개발자 대비 매우 경제적인 선택입니다. 예를 들어, 인도에서 채용하는 개발자의 평균 연봉은 다음과 같습니다.

- 0~2년차 주니어 개발자: 약 640만 원 ~ 1,300만 원
- 2~5년차 미들급 개발자: 약 1,300만 원 ~ 2,400만 원
- 5년 이상 시니어 개발자: 약 2,400만 원 ~ 3,200만 원

이는 한국에서 일반적으로 책정되는 개발자 연봉 대비 절반 이상 저렴한 수준으로, 중소기업에 상당한 비용 절감 효과를 제공합니다. 따라서 인도 개발자를 활용하는 것은 경제적 효율성을 높이는 효과적인 방법입니다.

글로벌
커뮤니케이션
능력

인도는 영어를 공용어로 사용하며, 이는 글로벌 비즈니스 환경에서 큰 강점으로 작용합니다. 인도의 소프트웨어 엔지니어들은 글로벌 기업과의 커뮤니케이션에 능숙하며, 이는 특히 한국과 같은 아시아 국가가 인도의 인재를 활용하는 데 유리하게 작용합니다. 원활한 영어 소통 능력은 한국 기업이 인도의 우수한 엔지니어와 협업하는 데 필수적인 요소입니다.

한국 기업을 위한 인도 인재의 이점



05

원격 협업 경험

코로나19 이후 인도의 많은 개발자들은 원격 근무 경험을 쌓아왔습니다. 이들은 Slack, Jira, Trello 등과 같은 협업 도구 사용에 익숙하며, 이러한 도구들은 원격 협업의 효율성을 높이는 데 중요한 역할을 합니다. 인도의 엔지니어들은 이러한 환경에 쉽게 적응하며, 한국 기업이 원격으로 이들과 협력할 때 높은 생산성을 유지할 수 있는 환경을 제공합니다.

06

글로벌 리더십 입증

인도 출신의 유명한 빅테크 CEO들은 인도의 기술 인재의 우수성을 입증해 왔습니다. 예를 들어, Google의 순다 피차이, Microsoft의 사티아 나델라, Adobe의 산타누 나라옌, IBM의 아빈드 크리슈나 등은 모두 인도에서 교육을 받고 글로벌 IT 업계에서 성공한 인물들입니다. 이들은 인도 기술 인재의 혁신성과 리더십을 보여주는 대표적 사례로, 인도의 소프트웨어 엔지니어들이 글로벌 무대에서 얼마나 중요한 역할을 할 수 있는지를 증명하고 있습니다.

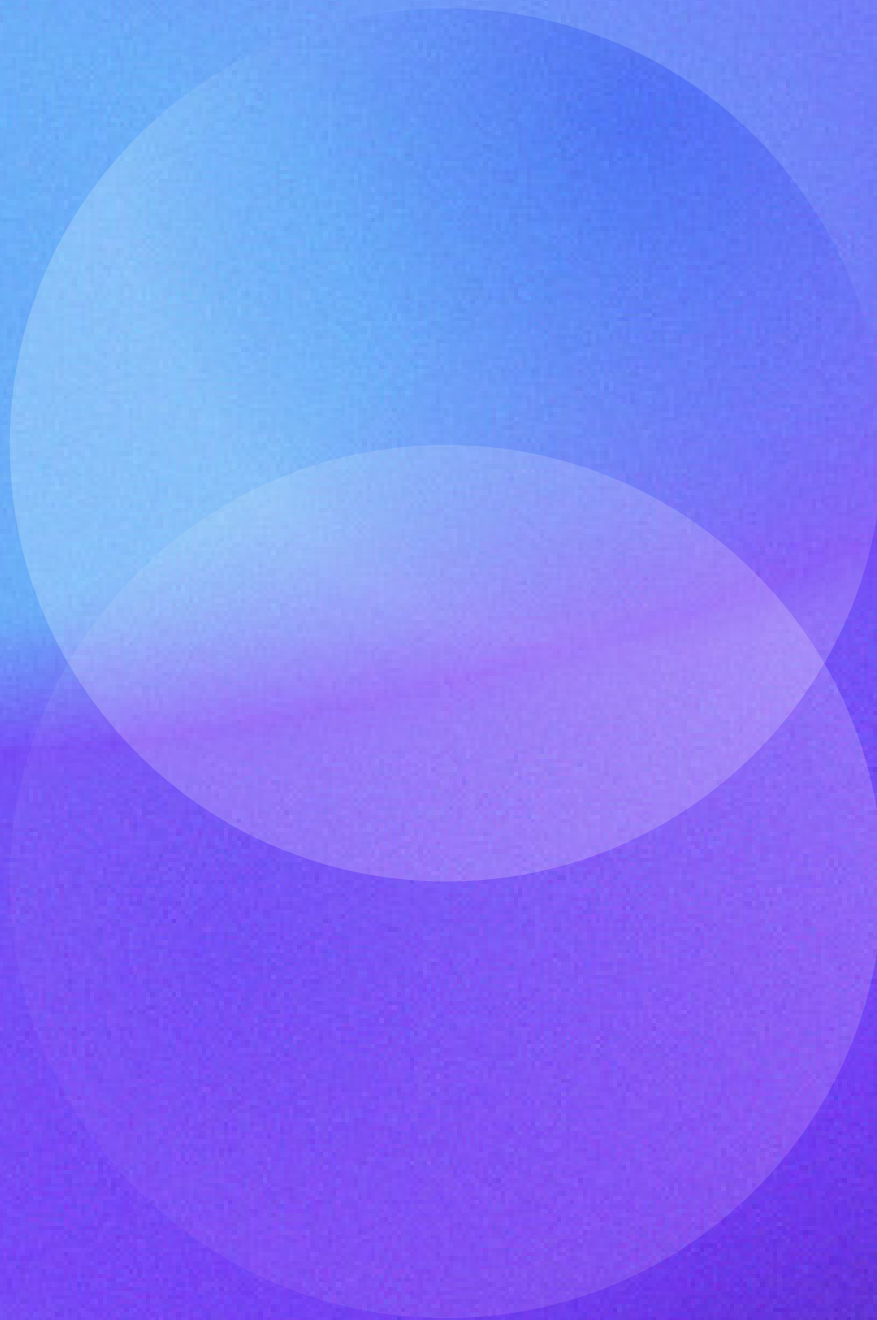
07

풍부한 인력 공급

인도는 약 1,000만 명에 달하는 소프트웨어 엔지니어 풀을 보유하고 있습니다. 이는 한국의 IT 인력 부족 문제를 해결할 수 있는 좋은 대안이 될 수 있으며, 중소기업들이 적합한 인재를 인도에서 충분히 찾을 수 있음을 의미합니다. 이러한 인재 풀을 활용하면 기업은 다양한 기술적 요구를 충족시킬 수 있는 맞춤형 인재를 채용할 수 있습니다.

PART 2

인도 인력 채용 방법



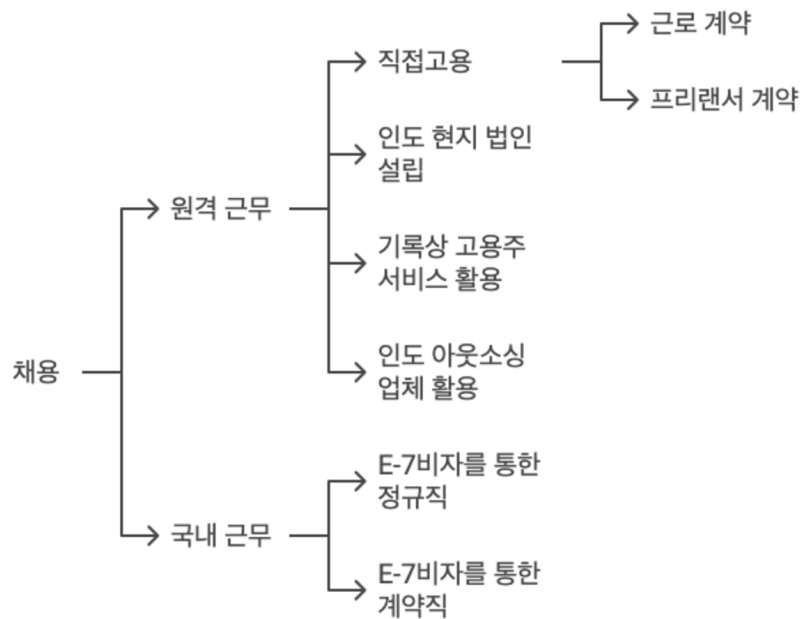
02

인도 인력 채용 방법

인도 개발자와의 협업은 크게 두가지 방식으로 일을 할 수 있습니다.

1. 원격 근무: 개발자가 인도에서 거주하면서 한국 회사 (또는 한국 내 프로젝트) 업무를 수행
2. 국내 근무: 개발자를 직접 한국 으로 초청(비자 발급)하여, 현장에서 함께 근무

각 방식 마다 고용·계약 구조, 비자·노동법,세무 이슈가 달라집니다. 아래에서는 가능한 방법들을 유형 별로 소개하고, 계약 구조 별로 설명하겠습니다.



01

인도 개발자를 '인도 현지'에서 원격으로 고용하는 방법

1.1 직접 고용

직접 고용 형태는 한국 회사가 인도에 거주하는 개발자와 별도의 법인(지사) 없이 근로계약을 직접 체결하는 형태입니다. 계약 형태는 근로 계약으로 하거나, 프리랜서 계약으로 진행이 가능합니다. 이 경우 개발자는 한국 회사의 '직원'이지만, 실제 근무지는 인도인 경우입니다. 월급 등 보수는 한국 회사에서 직접 송금하게 됩니다.

1.1.1 근로계약

■ 개념

- 회사(고용주)와 근로자(개발자) 간에 지휘·감독 관계가 존재하고, 회사가 임금·근로 조건을 결정하며, 개발자는 이에 종속되어 업무를 수행하는 형태.
- 통상적인 회사 직원 (정규직, 계약직 등)의 형태를 떠올리면 됨.

■ 특징

구분	내용
지휘·감독	회사가 근로시간, 업무 방식, 휴가, 복무 규정 등을 직접 지시하고 통제가능함.
근무 장소	원격일 수도 있으나, 근로자 입장에서는 '회사'의 지휘·감독을 받는 구조임.
임금	월급, 주급, 시급 등 정기성이 있고, 일정한 기본급을 지급하는 형태가 대부분.
노동법 적용	근무지가 인도라면 인도 노동법이 우선(최저임금, 퇴직금, 사회보험 등), 회사가 공식 고용주로서 법적 책임을 져야 함.
세금·사회보험	원칙적으로는 회사가 '원천징수', '사회보험 가입'등을 책임져야 하는 구조이지만, 해외 고용주가 인도에 법인이 없기 때문에 이행하기 어려움.
위장 고용 리스크	실제로는 근로자로서 상시 지휘·감독을 받고, 회사가 근태까지 전부 관리 한다면 "위장 프리랜서"로 간주될 위험이 존재

1.1.2 프리랜서 계약

■ 개념

- 회사와 개발자가 지휘·감독 관계가 아닌, 대등한 당사자로서 "업무 성과(프로젝트 납품 등)"를 제공하고, 그 대가를 지급받는 형태
- 일반적으로 도급(용역) 계약과 유사한 개념이며, "개발자가 독립적 사업자/사업체"로 활동한다는 전제를 갖음.

■ 특징

구분	내용
지휘·감독	회사는 개발자에게 근로시간·휴가·근태 등을 직접 통제하지 않고, 결과물 위주로 계약

대금 구조	프로젝트별 고정 금액, 작업 시간별 청구, 마일스톤마다 지불 등 다양한 방식.
노동법 적용	인도 노동법(최저임금, 퇴직금, 사회보험) 등 '근로자 보호' 규정이 적용되지 않는 것이 원칙
세금 처리	개발자가 개인 사업자로서 소득 신고를 수행하고, 회사와 직접 원천 징수 의무 관계는 감소.
세금(소득) 처리	인도 법인이 원천징수, 법인세, 부가세 등 직접 납부 및 회계처리
위장 고용 리스크	실제로는 근로자로서 상시 지휘·감독을 받고, 회사가 근태까지 전부 관리 한다면 "위장 프리랜서"로 간주될 위험이 존재

1.1.3 근로계약과 프리랜서 계약의 주요 차이

구분	근로계약	프리랜서
법적 지위	고용주(회사)와 근로자 간 종속적 관계	대등한 계약 당사자(사업자 대 발주처)
지휘·감독	회사가 근로시간, 업무방식, 휴가 등을 직접 통제	회사는 결과물에 대한 요구사항만 제시, 근무 방식 등은 개발자 재량
노동법 적용	근로기준법, 최저임금법, 사회보험 등 강행 규정 적용	원칙적으로 비적용, 예: 최저임금·퇴직금·연차 등 없음
급여(보수) 구조	통상 월급, 연봉, 시급 등 정기성 지급	프로젝트별, 기간별, 작업량별 등 변동성 큰 대금 지급 방식
세금(소득) 처리	고용주가 원천징수 후 차액 지급, 사회보험 부담	개발자 스스로 소득 신고, 회사는 도급비 명목 지급
사회보험 가입	고용주가 의무 가입	가입 의무 없음
위장 고용 리스크	근로계약이므로 위장 고용 이슈 없음, 단 노동법 위반 유의	지휘·감독이 심하면 "실질적 근로"로 보아 위장 고용 문제 발생 가능
목적	상대적으로 장기고용	프로젝트 성, 단기 계약

11.4 실제 직접 채용 시 유의 사항

■ 근로계약으로 채용할 경우

- “인도에 거주하는 근로자”에 대해 회사(해외 고용주)에게 적용되는 인도 노동법을 충분히 파악해야 합니다.
- 급여에서 원천징수 공제, 사회보험 가입 등 현지 법적 의무를 준수하기가 쉽지 않으므로, 장기적으로는 기록상 고용주 서비스나 현지 법인 설립을 고려해야 합니다.

■ 프리랜서 계약으로 채용할 경우

- 회사가 개발자의 근로 시간, 출퇴근, 휴가 등을 직접 통제하지 않아야 프리랜서 관계가 성립합니다.
- 만약 실무적으로 근로계약과 똑같이 지휘·감독한다면 “위장 고용”문제가 발생해, 인도 노동법이 적용될 수 있고, 추후 퇴직금·사회보장 분쟁이 생길 여지가 있습니다.
- 계약서 작성
 - (근로계약) “근무시간, 급여, 휴가, 4대 보험(혹은 인도 사회보험) 등”을 명확히 기재, 노동법 준수 조항을 포함해야 합니다.
 - (프리랜서) “프로젝트 범위, 대금 지급 방식, 납품 기한, 비밀유지, 지적재산권 귀속”등 성과물 중심으로 작성합니다.
 - 분쟁 대비를 위해, 준거법, 분쟁 해결 관할, 서명 후 교부 등을 반드시 명시합니다.

소득세 원천징수 문제

인도 법에서 일반적으로 “고용주가 급여 지급 시 원천징수”를 해야 합니다. 하지만 한국 회사가 인도 내 법인이 없다면, 실무적으로 원천징수를 하기 어려워서, 개발자가 직접 인도 국세청에 자진 신고를 해야 할 수 있습니다.

이때 회사와 근로자 간에 별도로 “개발자가 현지 세금 신고 및 납부를 직접 책임진다” 등의 합의가 필요합니다.

1.2 인도 현지 법인 설립

회사가 직접 인도 법인을 설립해, 법인이 개발자를 정식 고용하는 방식입니다. 법인 설립부터 세무·노무 관리까지 부담이 있지만, 안정적이고 장기적인 운영에 유리 합니다.

▪ 개념

- 한국 본사가 인도 현지에 법인(지사, 자회사 등)을 설립하고, 해당 법인이 개발자를 정식 고용하는 구조.
- 인도 법인이 자체적으로 근로계약을 체결하며, 급여·노무·세금·사회보험 등을 직접 관리.

▪ 특징

구분	내용
법적 지위	인도 법인이 개발자의 공식 고용주가 됨. 모회사(한국)는 '모회사-자회사' 관계로 지휘·관리 가능
지휘·감독	인도 법인이 근로자를 직접 관리(근무시간·휴가·급여 결정). 단, 실무 프로젝트 지시는 모회사와 협의 가능
노동법 적용	인도 노동법을 전면 준수해야 함(최저임금, 퇴직금 등). 인도 법인이 '법적 고용주'이므로 책임 명확
급여(보수) 구조	인도 법인이 개발자에게 월급/연봉 정기 지급, 내부 규정에 따라 상여·수당 부여 가능
세금(소득) 처리	인도 법인이 원천징수, 법인세, 부가세 등 직접 납부 및 회계처리
사회보험 가입	사회보험, 연금 등 인도 법령에 따른 의무 가입 의무. 인도 법인이 근로자 등록 후 부담금 납부
위장 고용 리스크	모회사-자회사 정상 구조이므로 '위장 채용'문제 없음. 인사·노무 실무는 인도 법인이 맡아 안정적 운영 가능
장단점	- 장점: 장기·대규모 인력 운영 시 법적 안정성 높음, 현지 시장 확장 용이 - 단점: 법인 설립·유지 비용, 회계·세무 관리 높음

1.3 기록상 고용주 서비스 활용

기록상 고용주 서비스를 활용해서 개발자를 채용할 수 있으며, 이 경우 기록상 고용주 서비스를 제공하는 업체가 공식 고용주가 되어 인도 노동법을 대행 준수합니다. 회사는 프로젝트 지시만 직접 담당하며, 법인 설립 없이도 합법적으로 인력을 확보할 수 있습니다.

▪ 개념

- 기록상 고용주란, 해외 회사를 대신해서 현지에서 개발자를 고용해주는 제 3의 서비스 업체.
- 개발자는 기록상 고용주 서비스 업체와 근로계약을 맺고, 실제 업무 지시·프로젝트 관리 등은 한국 회사가 담당.

▪ 특징

구분	내용
법적 지위	개발자의 공식 고용주는 기록상 고용주 업체(인도 현지 법인)이며, 한국 회사는 기록상 고용주 업체와 B2B 서비스 계약을 체결
지휘·감독	실무 업무(프로젝트, 코드 리뷰 등)는 한국 회사가 직접 지시 가능하나 근로조건(휴가, 급여 변경 등)은 기록상 고용주 회사가 최종 결정
노동법 적용	기록상 고용주 업체가 인도 노동법을 준수하므로, 한국 회사 자체는 별도 법적 책임이 크지 않음
급여(보수) 구조	한국 회사는 기록상 고용주 업체에게 서비스 비용과 실제 급여분을 일괄 지급하고 기록상 고용주는 개발자에게 월급(연봉)을 지급하고, 수수료를 공제
세금(소득) 처리	기록상 고용주가 개발자 소득세(원천징수)를 인도 세무당국에 납부하고, 한국 회사는 '용역비(서비스 비용)'로 회계 처리
사회보험 가입	기록상 고용주가 공식 고용주로서 연금, 사회 보험 가입·운영하고, 한국 회사는 별도 가입 의무 없음
위장 고용 리스크	한국 회사가 근태·휴가까지 전부 직접 통제하면, "실질 고용주" 논란이 가능하고, 역할 분담(업무 지시와 노무 관리)을 명확히 구분 필요.
장단점	<ul style="list-style-type: none"> - 장점: 법인 설립 없이 합법적으로 인도 인력 고용이 가능하고, 세무·노무 절차 대행으로 관리 편의성이 높음. - 기록상 고용주 서비스 수수료가 발생하고, 직접 고용 대비 비용이 증가함. 또한 기록상 고용주와 역할을 명확하게 구분 필요.

1.4 인도 아웃소싱 업체 활용

회사가 직접 개발자를 고용하지 않고, 인도 아웃소싱 업체(벤더)와 용역 계약을 맺어 서비스를 제공받는 방식입니다. 벤더사가 개발자 고용 및 관리(급여·휴가·근태)를 관리하고, 회사는 결과물과 일정·품질에 대해 요구하고 검토합니다.

▪ 개념

- 용역 계약을 통해, 벤더가 개발자를 고용·관리하고 회사는 프로젝트 결과물 위주로 협업.
- 회사는 근무조건 통제가 아닌 업무 요구사항·일정·품질만 벤더 측에 제시하고 개발자 관리는 벤더의 프로젝트 관리자가 수행.

▪ 특징

구분	내용
법적 지위	벤더가 개발자의 개발자의 공식 고용주이고 한국 회사는 발주처로서 결과물에 대한 요구 및 검수 권한을 가짐.
지휘·감독	벤더가 개발자의 근무시간·급여·휴가 등을 전담 관리하고, 한국 회사는 프로젝트 사양·일정·품질 요구사항을 벤더 측 프로젝트 관리자에게 전달, 벤더가 내부적으로 개발자를 배치·관리.
노동법 적용	인도 노동법은 벤더가 준수하고, 한국 회사는 노동법·사회보험 의무를 직접 부담하지 않음.
급여(보수) 구조	벤더가 개발자에게 월급/연봉을 정기 지급하고 벤더 내부 규정에 따라 상여·수당 등을 부여하고 한국 회사는 벤더와의 계약 대금(프로젝트 비용)을 지불.
세금(소득) 처리	벤더가 개발자 소득에 대해 원천징수, 법인세, 부가세 등을 직접 납부하고, 한국 회사는 벤더에게 용역 비용을 지급하여, 별도 세무 처리는 계약 형태에 따라 달라짐
사회보험 가입	벤더가 인도 노동법에 따른 연금, 사회 보험 가입·운영하고, 한국 회사는 별도 가입 의무 없음.
위장 고용 리스크	한국 회사가 개발자 근로조건까지 전면 통제하면, '실질 고용주'로 간주될 위험이 있고, 정상적인 용역 관계로 유지하려면, 지휘·감독은 벤더 프로젝트 관리자를 통해 간접적으로 이뤄져야 함.
장단점	<ul style="list-style-type: none"> - 장점: 직접 고용 부담이 없고, 초기 리소스 절감과 빠른 인력 투입 가능 - 단점: 회사가 세부 지휘·감독하기 어려워 품질·일정 관리에 신경 써야 함.

인도 개발자 채용 서비스

■ 슈퍼코더((주)세컨드팀)

전세계 기업들이 전세계의 뛰어난 개발자를 원격/현지 채용할 수 있도록 돕는 HR 플랫폼입니다. 기술력 검증과 함께 원격 근무 가이드라인을 제공하여 원활한 협업을 지원합니다.

■ 레드롭((주)맥킨리라이스)

아시아, 유럽, 미국 등지의 글로벌 인재 풀을 보유한 HR 플랫폼으로, 기술과 지역에 맞는 개발자를 한국 기업에 연결합니다. 팀 단위 채용 옵션과 전 과정 지원 시스템을 제공하여 다국적 프로젝트에 적합한 솔루션을 제공합니다.

02

인도 개발자를

'한국으로 초청'하여

국내에서 함께

근무하는 방법

인도 개발자를 국내로 초청해 현장 근무하게 할 수 있습니다. 보통 E-7(특정 활동) 비자를 통해 전문 인력을 유치하며, 체류 기간 동안 한국 법령(노동법, 4대 보험 등)을 준수 해야 합니다. 비자 발급, 숙소·정착 지원, 언어·문화 적응 등 준비가 필요합니다.

2.1 E-7 비자를 통해 정규직 또는 계약직으로 고용

■ 개념

- 전문 인력 유치를 위한 취업 비자로, IT개발자를 비롯한 특정 기술·전문 분야 근로자를 국내에서 고용가능하게 함.
- 한국 근로계약(정규직·계약직)에 기반하여, 회사가 비자 스폰서로서 고용제 의와 관련서류를 준비하고, 개발자는 E-7 비자를 받아 입국·근무.
- 체류 기간은 1~2년 단위로 발급하며, 고용 관계가 유지되면 갱신도 가능.

■ 특징

구분	내용
법적 지위	해외 인재가 E-7(특정활동) 비자를 받아 국내에 입국, 해당 회사 소속 근로자로 등록하면 회사는 고용주로서 출입국관리법 및 노동관계법을 함께 준수해야 함.

지휘·감독	회사는 근로계약을 통해 근무시간, 업무 범위, 휴가등을 정하고, 근로기준법 내에서 지휘·감독 권한을 행사하고 개발자는 회사 내 조직에 소속되어 정규 직원처럼 업무를 수행.
노동법 적용	한국 노동법에 적용을 받으며, 근로기준법상 부당해고, 근로조건 관련 조항 등 외국인 근로자도 동등한 보호를 받음.
급여(보수) 구조	회사가 국내 근로자와 유사한 월급/연봉 형태로 정기 임금을 지급.
세금(소득) 처리	근로소득세 원천징수, 연말정산 등 내국인과 동일한 절차를 따르고, 연봉 구간별 누진세율 적용 및 이중과세방지협정에 의해 일부 조정 가능.
사회보험 가입	4대 보험 가입 의무가 있고, 회사·근로자 각각 해당 보험료를 부담하며, 외국인 등록 후 정상적으로 국내 사회보장제도 적용.
위장 고용 리스크	실제 근로계약이 존재하므로, 도급·위장 고용 문제가 발생할 가능성을 낮으나, 근로조건(휴가, 퇴직금 등)을 명시하지 않거나 한국 노동법을 준수하지 않으면 분쟁이 생길 수 있음.
장단점	- 장점: 사내 협업·커뮤니케이션이 원활해지고, 국내 팀빌딩 효과가 올라감. - 단점: 비자 발급 절차(서류·행정)와 초기 정착(숙소, 문화 적응) 지원 부담, 장기 근속 실패 시 비자·체류 연장 문제 발생 가능

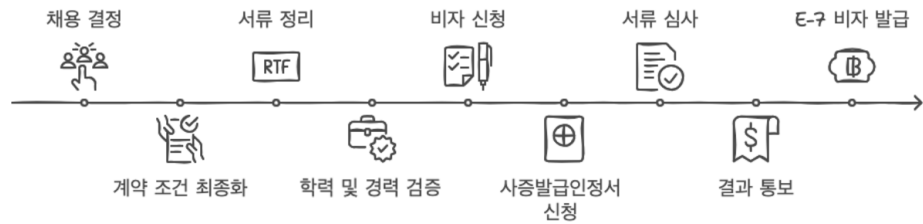
2.2 E-7 비자 발급 관련 가이드 라인

인도 개발자를 E-7(특정활동) 비자로 초청하여 국내에서 근무하게 할 때 필요한 절차 가이드라인 입니다. 실제 진행 시점에 따라 법규·행정 지침이 달라질 수 있으므로, 반드시 법무부 안내와 전문 행정사·노무사의 자문을 함께 참고하시기 바랍니다.

2.2.1 E-7 비자 개요

1. E-7(특정활동) 비자는 “대한민국 산업 발전에 필요한 전문인력”을 국내 사업장에서 고용하기 위해 발급 되는 취업비자입니다.
2. IT, 엔지니어링, 연구개발(R&D), 디자인, 해외마케팅 등 광범위한 분야에서 특정 자격 요건(학력·경력·연봉등)을 충족하면 신청 가능합니다.
3. 근무 장소가 대한민국 국내 사업장(본사·지사 등)이어야 하며, 고용주(한국 회사)와 근로계약을 맺는 형태입니다.

인도 개발자를 위한 한국 고용 절차



2.2.2 E-7 비자 발급 필요 요건

구분	내용
계약기간	<p>계약기간은 출입국관리법상 최소 90일 이상이어야 취업비자 발급 신청이 가능합니다.</p> <p>실무적으로 계약기간이 90일에 가까울 경우 비자 발급 가능성이 낮으므로, 최소 6개월 이상의 계약기간을 권장합니다.</p> <p>중도 퇴사 시에는 퇴사일로부터 15일 이내에 법무부 출입국에 중도 퇴사 신고를 해야 하며, 이를 통해 출입국관리법상 문제가 발생하지 않도록 합니다.</p>
연봉 요건	<p>2023년 기준 GNI: 44,0051,000원</p> <p>일반 연봉 기준: GNI의 80% 수준인 월 294만원 이상</p> <p>특별 연봉 기준: GNI의 70% 수준인 월 257만원 이상</p>
특별 연봉 기준 요건	<p>벤처기업 + 국내기업 취업 경력 3년 이내</p> <p>중소기업 + 국내기업 취업 경력 3년 이내</p> <p>중견기업법상 중견기업(비수도권) + 국내기업 취업 경력 3년 이내</p>
일반 요건	<p>도입직종과 연관성이 있는 분야의 석사 이상 학위 소지자</p> <p>도입직종과 연관성이 있는 학사 학위 소지자 + 1년 이상의 해당 분야 경력자</p> <p>도입직종과 연관성이 있는 분야에서 5년 이상의 근무 경력자 (인도 학위 및 경력 소지자는 해당국 아포스티유 또는 영사 확인 절차 필요)</p>
특별 요건	<p>세계 500대 기업에서 1년 이상 전문직종 근무 경력자</p> <p>세계 우수 대학 학사 학위 소지자</p> <p>국내 전문대학 졸업(예정)자</p> <p>국내 대학 졸업(예정) 학사 이상 학위 소지자</p>

2.2.3 주요 절차

1. 한국 회사의 사전 준비

- 채용 결정: 인도 개발자를 채용하기로 결정하고, 근로 계약서(임금, 근무조건, 계약기간 등)를 확정.
- 4대 보험, 회사 서류 정비: 회사가 “외국인 전문인력”을 고용할 수 있는 사업자등록, 납세·보험 가입 등 상태가 적법해야 함.
- 학력·경력 검증: 개발자의 전문성(학사 이상, 관련 경력 등)이 E-7 기준을 충족하는지 확인.

2. 비자 신청(출입국 당국 심사)

- 사증발급인정서(또는 사증발급허가) 신청: 회사(고용주)가 국내 관할 출입국·외국인청에 신청서를 제출.
- 서류 심사: 고용계약서, 회사 사업자등록증, 외국인 전문성 증빙(학위·경력증명), 필요 시 4대 보험 증빙 등.
- 결과 통보: 심사 후 사증발급인정서가 발급되면, 이를 인도 개발자에게 전달.

3. 주인도 한국 대사관/영사관 발급

- 인도 개발자는 사증발급인증서와 여권, 사진 등 구비서류를 가지고 주인도 한국대사관(또는 영사관)에 방문.
 - 최종 인터뷰/심사를 거쳐 E-7 비자 스티커를 여권에 부착받음
 - 비자 발급 후 한국에 입국 가능 (입국 시 공항에서 체류자격 부여)

2.2.4 비자 발급 서류 및 요건

- 사증발급인정신청서 (비자 신청서)
- 여권 사본 (유효기간 6개월 이상 - 인도 주재 대한민국 영사관에서 사증접수 시점 기준)
- 여권용 사진
- 고용계약서 원본 및 사본
 - 월 급여 단가, 일일 근무시간, 계약기간(3개월 이상), 근무내용 등 포함
- 사업자등록증 사본
- 법인등기부등본 등 설립기관 입증서류
- 대표자 신분증 사본 - 대표자 고용노동법 관련 범죄 이력이 있는 경우 초청 제

한, 출입국에서 대표자 인적사항으로 별도 조사

• 회사 관련 서류:

- 납부내역증명(납세사실증명) 또는 회사재무제표
- 납세증명서(국세완납증명서)
- 지방세 납세증명서
- 부가가치세 과세표준증명원
- 회사소개서

• 결핵검사서 (지정병원에서 3개월 이내 발급) - *결핵고위험 국가인 경우

• 학위증명서 또는 경력증명서 (해당 직종에 따라) - 인도 학위 및 경력인 경우
아포스티유 인증 또는 영사 확인 필수

• 자격증 사본 (해당 직종에 따라)

• 고용사유서

• 외국인투자기업등록증 (해당 시)

• 업종별 인/허가 서류 (해당 시)

• 외국인환자유치업등록증 (해당 시)

• 고용추천서 (해당 시)

• 수출·입 실적증명서 (해당 시)

• 한국어 능력급수 입증서류 (해당 시)

2.2.5 상세 절차 및 소요 시간

절차	설명	기간
기초 조사	해당 국가의 비자 요구 사항과 절차를 사전에 조사합니다.	-
진위여부 확인 작업	학위증 및 경력증명서 아포스티유 또는 영사 확인 절차를 진행합니다.	진행 기간 약 2~3주
서류 준비	필요한 서류를 준비하고 비자 신청서를 작성합니다.	-
국내 법무부 출입국 관리사무소 접수	법무부 출입국 사증발급인정신청서 접수	심사 기간 약 4주
사증발급인정서 발급	심사 후 발급됩니다.	-
인도 영사관 비자 접수	국내 발급된 사증발급인정서로 인도 주재 대한민국 영사관에서 비자 접수를 진행하고 사증발급확인서를 발급받습니다.	심사 기간 약 1~2주

외국인 등록	국내 입국 후 90일 이내 외국인등록 및 지문 등록 절차를 진행합니다.	심사 기간 약 3주
--------	---	---------------

Note 계약 해지 (해고) 시 제한

E-7 비자를 통해 국내에서 채용한 경우 한국 근로기준법을 준수해야 하며, 정당한 이유 없는 해고는 불법이 됩니다. 정규직의 경우, 사용자가 해고를 하려면 “근로기준법상 정당한 이유”가 있어야 하며, 해고 회피 노력이나 절차(사전 통보, 서면 통지 등)를 거쳐야 합니다. 계약직이라도 계약 기간 도중에 정당한 이유 없이 임의 해지하면, 부당해고로 인정될 수 있습니다.

사례 - 해외 개발자 E-7 비자 발급 절차 및 소요 비용

■ 내용

국내 기업 A사는 해외 개발자의 한국 내 근무를 위해 2024년 5월 7일 전문 비자(E-7) 신청 절차를 진행하였음. 비자 발급을 위해 해외 개발자가 준비해야 할 서류와 회사가 준비해야 할 서류를 명확히 정리하고, 국내 행정사 사무소에 비자 발급을 의뢰하였음.

■ 준비 서류

1. 해외 개발자 준비 서류:

- 경력증명서
- 여권 사진
- 여권 사본
- 사증발급인정 신청서

2. 회사 준비 서류:

- 사업자등록증
- 회사소개서
- 표준근로계약서 (일반기업 기준: 월 294만 원 이상)
- 고용사유서 (구체적인 활용 계획 포함)
- 납부내역증명 및 국세/지방세 완납증명서
- 부가가치세 과세표준증명원
- 4대보험 사업장 가입자 명부 및 고용보험 자격취득자 명부

- 등기사항전부증명서

- 위임장

▪ **업무 절차:**

1. 전담 행정사 배정 및 서류 준비 안내
2. 고객사 서류 준비 및 행정사의 1차 서류 검토 → 수정 보완 요청
3. 최종 서류 검토 및 원본 서류 전달
4. 출입국 비자 접수 및 인터뷰 진행 (약 3~4회 방문 필요)
5. 출입국 심사 후 비자 허가
6. 해외 재외공관에서 비자 발급
7. 외국인의 한국 입국 및 외국인등록 신청
8. 외국인등록증 전달

▪ **소요 기간:**

- 서류 준비 기간: 약 6주
- 출입국 심사 기간: 약 4주 (접수일로부터)
- 재외공관 비자 발급 기간: 약 2주 (신청일로부터)

▪ **비용:**

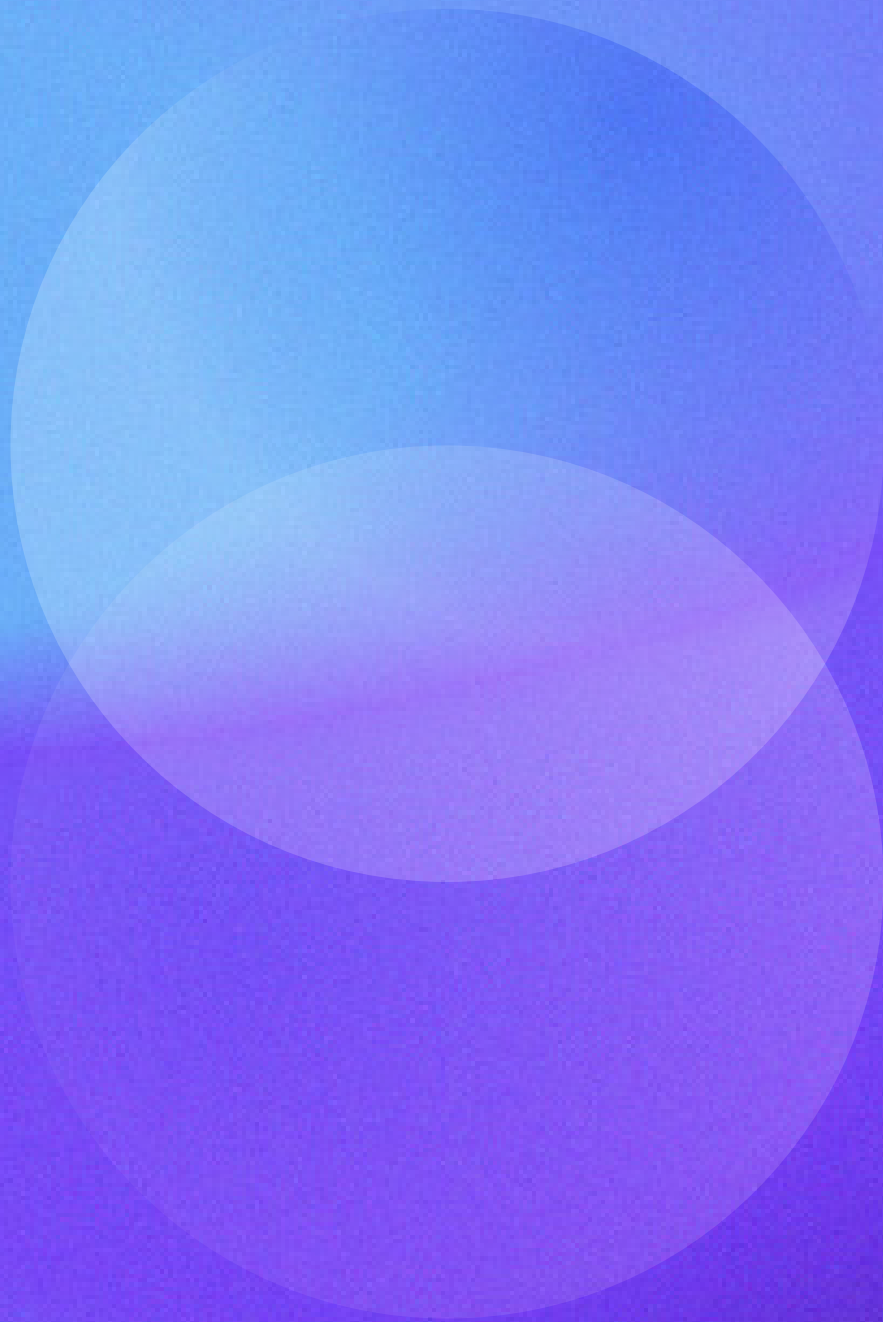
- 수임료: 85만 원
- 인지대: 3만 원
- 추가 서류 준비 비용: 아포스티유 비용

▪ **결과:**

2024년 5월 신청, 7월 비자가 허가되었고, 8월 인도 소재 대한민국 재외공관에서 비자를 발급받아 해외 개발자는 9월 한국에 입국하여 근무를 시작함. 비자 발급 과정에서 필요한 절차를 철저히 관리하고 서류 준비를 체계화함으로써, 비자 신청이 원활하게 이루어졌음.

PART 3

인도 개발자와
원격으로 일하는 방법



인도 개발자를 국내로 바로 초청해 함께 일하는 방식은 여러 가지 현실적인 어려움을 수반합니다. 회사 입장에서는 E-7 비자 발급 등 복잡한 행정 절차를 거쳐야 하고, 인재가 한국 생활에 안정적으로 적응할 수 있도록 숙소, 초기 정착 비용, 언어·문화 지원 등을 제공해야 하는 경우도 있습니다. 이러한 부분에서 추가 부담이 발생할 수밖에 없는데, 내부적으로 해외 인재 온보딩 프로그램이 제대로 마련되지 않은 회사라면 정착 과정에서 큰 문제가 생길 위험이 큼니다.

개발자 입장에서도 본국에서의 생활을 단기간 내에 정리하고 새로운 환경인 한국으로 이주한다는 것은 쉽지 않은 결정입니다. 직장뿐 아니라 가족, 주거, 기존 커뮤니티 등 개인적 환경을 완전히 바꿔야 하고, 언어나 문화 차이로 인한 심리적 부담도 상당합니다. 만약 회사가 계약 기간을 짧게 제안하거나 장기적인 비전이 불확실할 경우, 개발자로서는 이러한 위험을 감수하면서까지 이주를 선택하기가 더욱 어려워집니다.

결국 개발자와 원격 근무를 먼저 시도하는 것이 회사와 개발자에게 리스크를 최소화할 수 있는 방안입니다. 초기에는 비대면으로 협업하며 성과와 적응 가능성을 살펴보고, 이후 필요에 따라 온사이트 근무를 검토하고, 원격 근무가 어느 정도 안정화된 뒤에, 더 장기적으로 한국에서 일할 의향을 가진 개발자가 비자를 신청하고 이주를 고려 하는 것이 좋습니다.

코로나 이후 원격 근무는 급격히 확산되면서 많은 회사가 한 번쯤은 도입해 본 방식이 되었습니다. 하지만, 여전히 원격 근무를 경험하지 못한 회사도 많고, 비대면 커뮤니케이션에 대한 경험이 부족한 곳은 여전히 시행착오를 겪고 있습니다. 특히 시차나 언어, 문화 차이 같은 글로벌 환경에서 오는 문제를 한 번도 경험해보지 못한 조직은 원격 근무 효율이 기대만큼 나오지 않아 혼란을 겪을 가능성이 높습니다.

해당 챕터는 인도 개발자와 원격 근무를 하려는 기업을 위해 작성되었습니다. 원활한 커뮤니케이션 방법부터 시차 조율, 문화적 이해, 협업 툴 활용, 보안 이슈 관리, 그리고 계약 체결 시 고려해야 할 사항들까지 폭넓게 살펴보면서, 원격 근무가 가져올 수 있는 긍정적인 성과를 극대화하는 방법을 구체적으로 안내하고자 합니다.

01

원격 근무 도전 과제

원격 근무는 공간의 제약을 극복한다는 점에서 매력적이지만, 조직과 개인 모두에게 새로운 형태의 도전 과제를 안겨줍니다. 먼저, 서로 다른 시간대에서 일하는 구성원 간에 “언제 실시간으로 연락을 주고받을 것인지”에 대한 기본적인 문제가 발생합니다. 특히 글로벌 팀의 경우 일부 시간대에만 겹치는 근무 시간이 제한되어, 즉각적 의사소통이 어려워지거나 화상회의 시간을 잡기 위해 누군가는 새벽이나 늦은 밤에 참여해야 하는 부담이 생길 수도 있습니다.

의사소통 자체도 대면 환경에 비해 훨씬 복잡해집니다. 이메일이나 메신저, 화상회의 같은 비대면 채널을 통해 업무를 진행하다 보면, 사소한 뉘앙스나 맥락이 충분히 전달되지 못해 협업 속도가 느려지거나 오해가 생길 수 있습니다. 특히 다국적 팀에서는 언어 장벽과 문화적 차이까지 겹치기 때문에, 의도치 않은 갈등이 빚어질 가능성이 커집니다.

원격 근무의 도전 과제



많은 회사 입장에서는 원격 근무로 협업하는데 있어서 원격 근무 프로세스를 만들거나 협업 도구들을 세팅하는 것에 경험이 없거나 잘하지 못하는 경우가 많습니다. 기존의 업무 방식이 '사무실 근무' 기반 이고, 원격근무가 처음이라면 채택·원격 근무자가 제시간에 일을 하고 있는지, 프로젝트 진행 상황은 어떤지 파악하기가 쉽지 않습니다. 생산성 측정 기준이나 성과 평가는 기존과는 다른 모델이 필요해지고, 각종 협업 툴을 어떻게 도입하고 정착시키는 것이 쉽지 않을 수 있습니다. 또한 직원들의 다양한 원격 근무 환경에서 일하므로, 보안이나 데이터 보호 측면에서 평소보다 더 세심한 정책과 기술적 조치가 필요합니다.

02

원격 근무 가이드 베스트 프랙티스

- ☐ 전기·통신 인프라 준비
- ☐ 하드웨어 제공
- ☐ 원격 근무 시간 확립
- ☐ 커뮤니케이션 시스템 구축
- ☐ 보안 관리

2.1 안정적인 전기·통신 인프라 준비

인도의 경우 한국 만큼 안정된 전기·통신 인프라를 가지고 있지 않습니다. 따라서 인도 개발자와 원격 협업을 시작하려면, 개발자가 일하는 작업 공간의 전력 공급 상황과 인터넷 안정성, 협업 툴 사용 가능 여부 등을 꼼꼼히 확인해야 합니다. 개발자는 UPS, 백업 인터넷, 보안 툴 세팅 등 필요한 장비와 환경을 미리 갖춰야, 협업 과정에서 불필요한 중단이나 업무 지연을 최소화할 수 있습니다.

2.1.1 체크 리스트

	체크 사항
정전 빈도	개발자가 업무하는 지역에서 전력 공급이 얼마나 자주 끊기는지, 평균 정전 시간이 얼마나 되는지 확인이 필요합니다.
UPS(무정전 전원 장치) 여부	개발자의 PC나 인터넷 장치가 꺼지지 않도록 UPS가 설치돼 있는지, 또는 설치 계획이 있는지 파악하는 것이 중요합니다.
인터넷 회선 종류	광케이블, DSL, 모바일 테더링 등 인터넷에 연결 가능한 방식을 확인하여 인터넷 인프라를 확인합니다.

인터넷 속도·지연	개발 과정에서 화상회의 또는 화상회의를 할 경우 지연이 없는지 인터넷 속도를 확인합니다.
백업 통신망	주 인터넷이 끊겼을 때, 보조 인터넷 또는 모바일 핫스팟 등 대체 연결 방법이 마련돼 있는지를 확인해서 접속이 끊기는 부분을 최소화 합니다.

2.1.2 추천 사항

	추천 사항
UPS 또는 대체 전력 솔루션	전력 인프라 불안정한 지역에서 개발자가 안정적으로 원격 근무를 하기 위해서는, UPS나 휴대용 전력 장치 등 대체 전력 솔루션을 미리 준비해 업무 중단을 최소화하는 것이 중요합니다. 가벼운 정전이 가끔씩 발생한다면 오프라인/라인 인터랙티브 UPS로 짧은 시간 동안 업무 연속성을 보장하도록 하고, 정전이 빈번하고 장시간 지속된다면, 대형 UPS 또는 소형 발전기, 보조 배터리 등을 함께 고려해서 장시간 전력을 확보할 수 있는 방안을 마련하도록 해야 합니다.
이중 인터넷 라인	이중 인터넷 라인을 구성할 때는 가능하면 서로 다른 통신사를 사용하는 것이 이상적입니다. 하나의 통신사에서 유선과 무선 서비스를 모두 제공받을 수도 있지만, 이 경우 같은 통신망 장애나 같은 기지국·백본망 문제로 인해 두 라인이 동시에 끊길 수 있습니다.

2.2 개발자에게 하드웨어 제공

원격 근무 개발자의 하드웨어(노트북 등) 준비는 회사가 직접 장비를 사서 보내는 방식, 일정 금액을 지급해 개발자가 구매하는 방식등 다양하게 운영될 수 있습니다. 회사의 보안정책, 예산 규모, 배송·통관환경, 개발자 선호 등을 종합하여 선택할 수 있습니다.

2.2.1 체크리스트

	체크 항목
비용	비용을 회사와 개인이 각각 어느 비율로 부담할지 결정해야 합니다.

회사 표준	OS·보안·사양의 일관성이 중요한지도 고려해야 합니다.
지역·통관	인도의 경우 해외 배송과 통관이 까다로워 현지 구매 방식이 수월 할 수 있습니다.
보안	민감한 프로젝트라면, 회사에서 완전히 세팅한 장비를 지급하는 것이 안전합니다.
개발자 필요	개발가 선호하는 OS/플랫폼이 있다면, 일정 예산 범위 내에서 '장비 선택권'을 주는 식의 혼합모델도 가능합니다.
장비 수명 관리	개발 장비는 2~3년 주기로 성능 저하나 유지보수 이슈가 생길 수 있으므로, 교체·업그레이드 주기도 미리 설정해 놓는 것이 좋습니다.

2.2.2 노트북 지원 방식

지원방식	설명	장점	단점	운영 시 유의 사항
회사가 직접 구매	회사가 노트북을 직접 구매하여 개발자에게 배송	회사유휴노트북 사용가능하고보안설정가능	해외 배송 및 통관 비용과 시간 소요	반납 정책을 정해야 합니다.
장비 예산 지급	회사가 일정금액의 장비예산지급	해외 배송 부담 감소	회사 자산 분류 어려움	보안 가이드를 정해야 합니다.
개인 노트북 사용	개발자가 자신의 기존 노트북을 사용	회사 비용 최소화	업무에 필요한 사양 확보 어려움	최소 기준 사양 안내 및 보안 요구 사항 준수
노트북 임대	현지 렌탈 서비스를 통해 노트북을 일정기간 임대	초기 비용 부담 적고, 유지보수도 업체 부담	장기적으로비용이 높고, 데이터 유출의 위험이 커짐	보안설정적용 및 장비 반납 및 데이터 삭제 절차 마련

2.2.3 추천 사항

	추천 사항
장비 관리 정책	노트북 지급 시, 장비 지급·관리 정책을 서면으로 공유하고, 서명을 받아 놓습니다. 서면에 파손·분실·반납·수리 등 상황에 대해 구체적으로 기술해 두면, 분쟁 발생 시 불필요한 갈등을 줄일 수 있습니다.

보안 설정·원격 관리 툴	회사가 장비에 MDM 또는 원격 삭제/트래킹 기능을 세팅해 두면, 분실 시 기밀 데이터 유출 위험을 줄일 수 있습니다. 또한 암호화를 활성화해, 물리적으로 장비가 유출되더라도 데이터가 쉽게 노출되지 않도록 대비할 수 있습니다.
퇴사(계약 종료) 프로세스	계약 만료가 임박했거나 해지 통보가 나온 시점에, 장비 반환 절차와 데이터 백업, 회사 데이터 삭제를 안내 하고, 반환 기간 내 반납이 이뤄지지 않으면, 회사가 취할 수 있는 법적 조치를 정의해 둡니다.
보험·비용 처리	고가 장비는 회사가 보험에 가입하거나, 개발자와 일정 비용 부담 모델을 운영하면 리스크를 완화할 수 있습니다. 파손 시 경미한 손상은 회사가 부담하지만, 심각한 파손 시 개발자 책임 분담을 문서화하여 과실 판단을 명확히 합니다.

2.2.4 사례 공유

▪ A사 - 노트북 임대

A사의 경우에 노트북 구매에 대한 높은 비용도 부담이 되고, 개발자에게 장비 유지보수 및 회수에 대한 어려움도 고려해서 인도 렌탈 업체에서 임대를 결정하였습니다. 임대를 할 경우에 노트북 사양을 정하고, 기간(1개월, 3개월, 6개월, 12개월)을 정하게 되는데, 업무 초기에는 3개월 계약으로 시작하고 이후에 장기간 계약으로 변경 하였습니다. 임대 시 월 비용과 보증금의 비용이 발생하는데, 월 비용의 경우 개발자가 비용처리하고, 보증금의 경우에 현지 개발자가 보증금을 내는 방식으로, 향후에 장비 반납 시 개발자가 보증금을 회수해 가는 방식으로 복잡한 행정처리를 간소화 하였습니다.

▪ B사 - 노트북 지원금 분할 지급

AB사의 경우 개발자가 개인 노트북을 사용하였는데, 노트북 사양이 업무하는데 있어서 효율이 저하되어서, 노트북 구매 지원을 요청하였습니다. 하지만 B사의 경우에 노트북 지원을 한번에 해주는 것에 부담 때문에, 개발자가 선구매 하고, 24개월에 걸쳐 비용을 처리해주는 것으로 합의 하였습니다. 이를 통해, 개발자의 경우에는 고사양의 노트북을 활용해서 업무 효율성을 높이고, 회사의 경우에는 높은 비용에 대한 부담을 덜고, 개발자의 노트북 회수 절차에 대한 어려움도 해결할 수 있었습니다. (이 경우 개발자의 개인 노트북 구매를 지원한 경우로, 노트북은 개발자가 소유하게 됩니다.)

▪ C사 - 노트북 지원금 일시 지급

C사의 경우 개발자와 2년간 협업을 해오면서 개발자에 대한 신뢰가 높은 상태였습니다. 고객사의 경우에 개발자 노트북 사양이 업무에 영향을 주면서, 개발자의 노트북 구매를 100% 한번에 지원하였습니다. 이를 통해 개발자의 경우, 높은 금전적인 부담을 줄일 수 있었습니다. (이 경우 노트북은 회사 소유로 회수에 대한 부분을 사전에 협의 하였습니다.)

2.3 원격 근무 시간 확립

해외 개발자와 원격 협업을 성공적으로 수행하기 위해서는 업무 시간, 휴일을 명확히 정의하고 체계적으로 관리하는 것이 중요합니다. 시차와 문화적 차이를 고려한 유연한 업무 시간 설정, 각 국가의 공휴일을 공유하는 것이 중요합니다.

2.3.1 업무 시간 정의

인도 개발자와 원활한 협업을 위해서는 시차 관리, 인도 휴일등을 사전에 커뮤니케이션하고 일하는 것이 중요합니다.

2.3.2 시차 계산

한국과 인도의 시간 차이는 3시간 30분 입니다. 인도가 한국 보다 3시간 30분 늦습니다. 즉, 인도 오전 9시는 한국 시간 오후 12:30분 입니다.

한국	UTC+09:00
인도	UTC+5:30

2.3.3 겹치는 시간 정의 (코어 시간)

인도와 한국의 근무 시간을 고려하여, 가능한 겹치는 근무 시간을 설정합니다. 최소한 3시간 이상 근무 시간이 겹칠 수 있도록 코어시간을 정의 합니다.

한국 표준 근무 시간	오전 9시 ~ 오후 6시 (점심시간 오후 12시 ~ 오후 1시)
인도 표준 근무 시간	오전 9시 ~ 오후 6시 (점심시간 오후 1시 30분 ~ 오후 2시 30분)
겹치는 시간대 예시	한국 오후 1시 ~ 오후 4시 (인도 오전 10시 30분 ~ 오후 1시 30분)

사례 공유 - A사 개발자 근무 시간 조정 (탄력 근무제)

A사는 한국팀의 경우에 오전 10시에 출근하고, 개발자는 인도 시간 아침 7시에 업무에 시작하도록 맞추고 있습니다. 즉 한국 시간 10시 30분부터 협업이 가능하고, 개발자는 한국 시간 7시 30분에 업무를 종료 하고 있습니다. 이를 통해 겹치는 시간을 7시간 30분까지 확보해서, 시차로 인한 어려움 없이 협업을 하고 있습니다.

2.3.4 휴일 관리

인도 개발자와 협업 시 인도 공휴일 일정을 사전에 파악하고 있어야 합니다. 공휴일이 포함된 기간 동안 중요한 마일 스톤이나 데드라인을 피하고, 필요한 경우 일정을 미리 조정해야 합니다. 특별히, 인도는 각 주마다 추가적인 공휴일이 있을 수 있으므로, 사전에 신청 절차를 마련해서 업무에 차질이 없게 해야 합니다.

▪ 2024년 인도 공휴일 (델리)

날짜	공휴일 이름
1월 26일	Republic Day
2월 26일	Maha Shivaratri
3월 14일	Holi
3월 31일	Idul Fitr
4월 10일	Mahavir Jayanti
4월 18일	Good Friday
5월 12일	Buddha Purnima
6월 7일	Bakrid / Eid al Adha
7월 6일	Muharram
8월 15일	Independence Day
8월 16일	Janmashtami
9월 5일	Eid e Milad
10월 2일	Vijaya Dashami
10월 2일	Gandhi Jayanti
10월 20일	Diwali

2.4 개발자와 커뮤니케이션

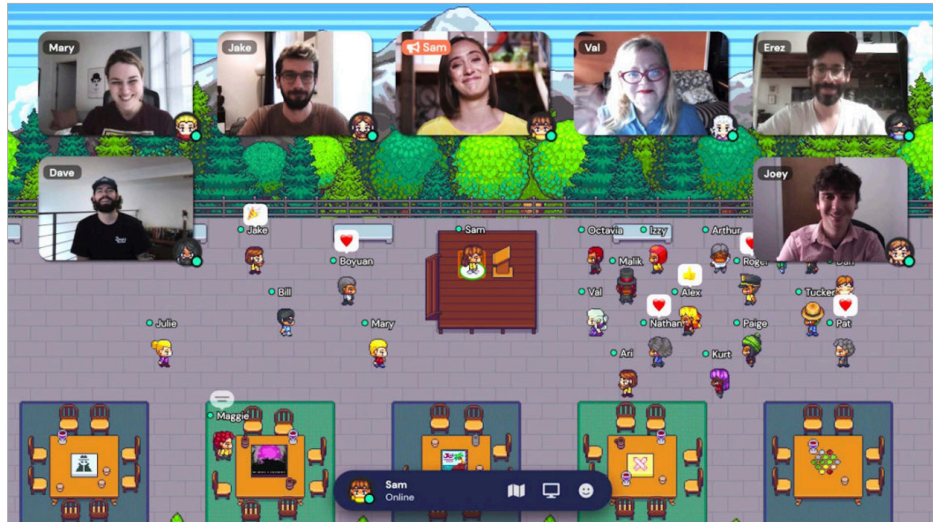
인도 개발자와 원격으로 협업할 때 효과적인 커뮤니케이션은 프로젝트의 성공과 팀의 효율성을 좌우하는 핵심 요소입니다. 다양한 문화적 배경과 시차, 언어 차이 등을 고려할 때, 원활한 의사소통은 필수적입니다.

2.4.1 커뮤니케이션 종류

커뮤니케이션 종류	적용 시기	장점	단점
회사가 직접 구매	실시간 논의가 필요할 때, 긴급한 문제 해결 시, 정기 회의 시	즉각적인 피드백 가능, 빠른 의사 결정 지원, 팀원 간의 유대감 형성	일정 조율이 어려움
비동기식 커뮤니케이션	시차가 큰 팀과 협업 시, 정보 공유 및 업데이트 시	시간에 제한 받지 않고 가능 기록이 남아 추후 참고 용이 팀원들이 자신의 일정에 맞춰 응답 가능	즉각적인 피드백이 어려움
서면 커뮤니케이션	공식 문서화가 필요할 때, 상세 정보 전달 시	명확하고 구체적인 정보 전달, 기록이 남아 추후 참조 가능 오해 최소화	작성에 시간이 많이 소요될 수 있음
구두 커뮤니케이션	긴급한 문제 해결 시 복잡한 이슈 논의 시	빠른 정보 정보 교환 가능, 즉각적인 질문과 답변 가능, 복잡한 논의에 유리	기록이 남지 않아 후속 관리가 어려움, 일정 조율이 어려움
영상 통화	얼굴을 보며 소통이 필요할 때, 비언어적 신호를 전달할 때	비언어적 신호를 통해 의사소통 효과 증대, 팀원 간의 유대감 형성, 복잡한 논의에 유리	안정적인 인터넷 연결 필요, 일정 조율이 어려움
메세징 앱	빠른 질문이나 간단한 정보 공유 시	실시간 대화 가능 다양한 채널을 통한 주제별 소통, 검색 기능을 통해 이전 대화 쉽게 찾기	과도한 메세지로 인한 정보 과부하, 비공식적인 대화로 인한 오해 가능성
이메일	공식적인 의사소통이 필요할 때, 상세한 정보 전달 시	공식적인 기록 남김 비동기적으로 소통 가능, 상세한 정보 전달 가능	응답 시간이 길어질 수 있음, 이메일을 보지 못하는 경우도 있음

2.4.2 커뮤니케이션 툴

커뮤니케이션 종류	툴	장점	단점
동기식 커뮤니케이션 (영상통화)	Zoom	고화질 영상 및 음성 화면 공유 및 녹화 기능, 웹 세미나 지원	무료 버전의 시간 제한 존재
	Microsoft Teams	채팅과 통합, 영상회의, 파일 공유, Office 365 와의 연동	복잡한 인터페이스
	Google Meet	Google Workspace와 의 연동, 간편한 사용성	고급 기능이 제한적 대규모 회의 시 성능 저하
	Gathertown	가상 오피스 환경으로 자연스러운 소통 가능, 실제 물리적 오피스와 비슷하게 사용자에게 접 근 하면 소통 가능	컴퓨터 리소스를 많이 소모해서 컴퓨터 속도가 느려짐
비동기식 커뮤니케이션 (메세징앱)	Email (Gmail, Outlook 등)	공식 기록으로 남김 비동기적 소통 가능 상세한 정보 전달 가능	느린 응답 정보의 추적이 어려움
	Slack	다양한 채널을 통한 주 제별 소통, 검색 기능을 통해 이전 대화 쉽게 검 색 가능	과도한 메시지로 정보 과부하 채널관리의 어려움
	Microsoft Teams	팀 내 실시간 채팅과 파 일 공유, 다양한 어플리 케이션과 통합	메시지 관리가 어려움 알림 설정이 복잡함
서면 커뮤니케이션	Google Docs	실시간 공동 편집 가능 댓글 및 수정 추적 기능	대규모 문서 관리가 어려 움
	Notion	노트작성, 데이터베이스 기 능 제공, 다양한 템플릿 제공	초기 설정에 학습이 필 요, 복잡한 프로젝트에 서는 관리가 어려움
	Confluence	개발 프로젝트에 최적화 Jira와의 통합	사용자 인터페이스가 복 잡함



<Gather Town>

2.4.3 커뮤니케이션 방법 및 프로세스 예시

■ 커뮤니케이션 도구 및 용도 정의

도구	용도	응답 예상 시간
Slack	일상적인 소통, 빠른 질문 및 답변	즉시 ~ 1시간 이내
Zoom/Google	정기적인 화상회의, 팀 미팅	회의 예약 전까지 응답 필요
Jira/Asana	프로젝트 관리, 작업 할당 및 추적	업무 요청시 24시간 이내 응답
Confluence/ Notion	문서화, 지식 공유	비동기적 응답 (필요 시 수일 내 응답)
Email	공식적인 커뮤니케이션, 장기적 인 기록	24시간 이내

■ 정기 커뮤니케이션 스케줄

회의 유형	빈도	목적
데일리 스탠드업	매일	당일 업무 계획 공유, 장애물 파악
주간 스프린트 계획	매주 또는 격주	스프린트 목표 설정, 작업 분배
스프린트 리뷰	스프린트 종료 시	완료된 작업 검토, 피드백 수집
스프린트 회고	스프린트 종료 시	프로세스 개선점 도출, 팀의 협업 방식 평가
월간 팀 미팅	매월	장기 목표 설정, 전반적인 팀 상태 점검

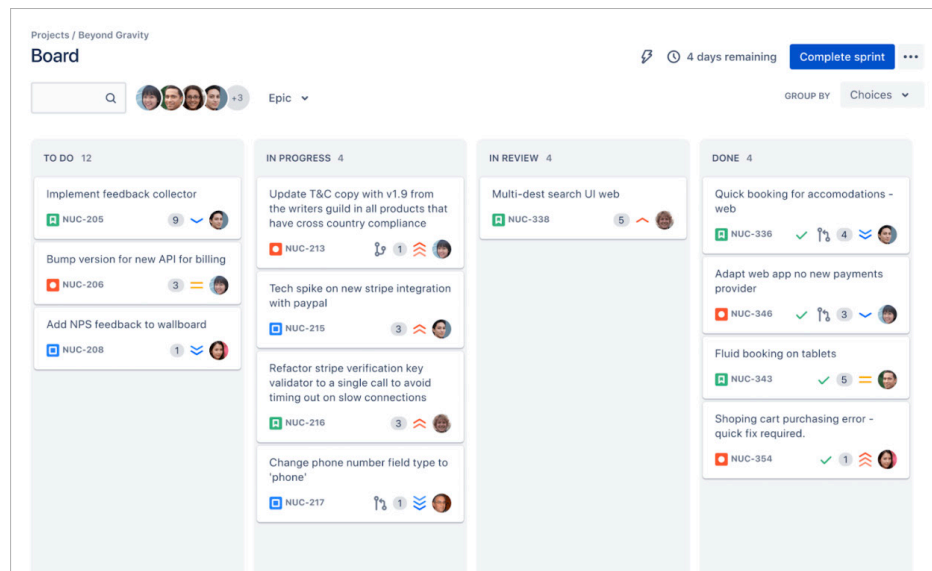
■ 비동기 커뮤니케이션 절차

업무	설명
업무 보고	매일 또는 매주 업무 보고서를 작성하여 Jira/Asana에 업로드 합니다.
이슈 공유	Slack이나 Jira를 통해 이슈를 공유하고, 비동기적으로 해결 방안을 논의 합니다.
문서 공유	Confluence/Notion을 통해 프로젝트 관련 문서를 중앙화하여 언제든지 검색 및 접근 가능하게 합니다.

■ 개발 프로젝트 시 스프린트 동안 커뮤니케이션

스프린트 단계	시간	활용 툴	활용 방법	
스프린트 계획	스프린트 시작 전	Jira, Confluence, Slack	Jira: 스프린트 백로그 생성 및 이슈 할당 Confluence: 스프린트 목표 및 요구사항 문서화 Slack: 초기 계획 공유 및 질의 응답	스프린트의 목표와 범위를 명확히 설정 팀원 간 역할 분담 및 기대치 공유 문서와 이슈의 중앙 관리
일일 스탠드업	매일 아침 (정해진 시간)	Jira, Slack	Jira: 각 팀원의 진행 상황 업데이트 Slack: 스탠드업 채널에서 간략한 상태 공유 및 이슈 토론	팀의 현재 진행 상황 파악, 장애물 신속 식별 및 해결, 팀원 간의 지속적인 소통 유지
작업 수행	스프린트 중반	Jira, Confluence, Slack	Jira: 태스크 진행 상황 업데이트, 이슈 트래킹 Confluence: 기술 문서, 코드 리뷰 가이드, 설계문서작성 Slack: 실시간 협업, 질문 및 답변, 파일 공유	작업 관리 및 추적 필요한 정보와 문서 중앙화 문제 공유와 커뮤니케이션
협업 문서화	스프린트 중반~후반	Confluence, Slack	Confluence: 회의 노트, 결정사항, 기술 문서 업데이트 Slack: 프로젝트 관련 채널에서 정보 공유 및 피드백 수집	프로젝트 관련 모든 정보를 문서화 팀원간 지식 공유

스프린트 리뷰	스프린트 종료 전	Jira, Confluence, Slack	Jira: 완료된 이슈 및 작업 검토 Confluence: 스프린트 결과 문서화 Slack: 리뷰 회의 일정 공유 및 피드백 논의	스프린트 결과물 검토 및 평가 고객 및 이해 관계자와의 피드백 공유 개선점 도출
회고	스프린트 종료 후	Confluence, Slack	Confluence: 회고 노트 작성 Slack: 회고 회의 알림 및 회의 후 피드백 공유	팀의 프로세스 및 협업 방식 개선 다음 스프린트에 적용할 액션 아이템 도출
스프린트 종료	스프린트 종료 후	Jira, Confluence, Slack	Jira: 완료된 스프린트 이슈 아카이브 및 다음 스프린트 계획 Confluence: 회고 결과 반영 Slack: 다음 스프린트 계획 공유 및 준비	스프린트 종료 후 정리 및 다음 스프린트 준비 지속적인 개선을 위한 정보 반영



<Atlassian사의 Jira>

사례 - 원격 협업 시스템 도입을 통한 업무 효율화

■ 문제 상황

국내 회사 A는 체계화된 IT 협업 툴 없이 카카오톡과 이메일만으로 해외 개발자와 협업을 시도함. 다양한 주제를 동시에 논의하다 보니 메시지가 뒤엉

켜 업무내용이 파악되지 않았고, 긴급한 논의 시 국제전화나 카카오톡 보이스톡을 이용해야 하는 불편함이 발생함. 화상미팅 역시 예약과 초대 과정이 번거로웠음.

■ 문제 원인 파악

회사는 동기적 및 비동기적 커뮤니케이션 시스템이 모두 부족한 상황으로, 효율적인 협업을 위한 IT 툴이 전혀 도입되지 않은 상태였음.

■ 대처 방안

• 비동기적 커뮤니케이션 강화:

- Slack 도입: 부서별, 프로젝트별, 주제별 그룹을 나누어 체계적으로 소통하도록 함. 대화와 파일이 클라우드에 영구적으로 저장되어 정보 검색과 파일 공유를 손쉽게 진행함.

- 업무 관리 툴 도입: 프로젝트 진행 상황과 코드 관리를 위해 Jira, Confluence, GitHub를 세팅하여 업무를 일목요연하게 정리하고 이해관계자 간 정보를 체계적으로 공유함.

• 동기적 커뮤니케이션 개선:

- Gathertown 활용: 무료 가상오피스를 도입하여 즉각적으로 화상 미팅과 화면 공유를 할 수 있도록 함. 긴급하고 깊이 있는 논의가 필요한 상황에서 높은 활용성을 제공함.

■ 결과

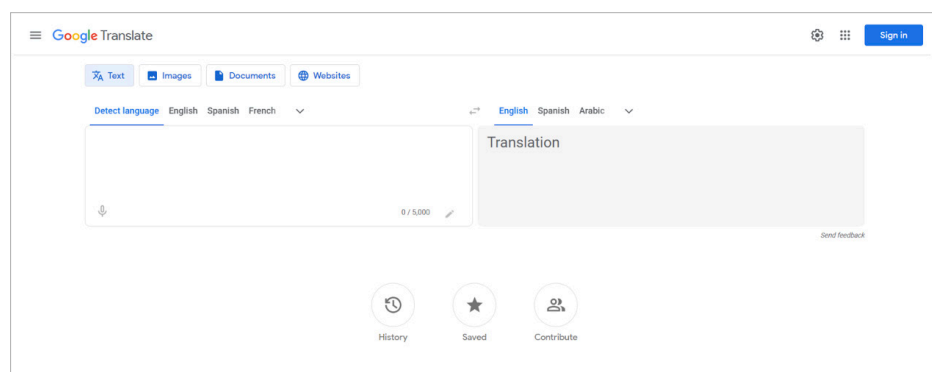
단 일주일 만에 해외 개발자와의 원격 협업 시스템을 체계적으로 구축하였으며, 생산성이 크게 향상되었음. 해외 팀원과의 소통뿐만 아니라 국내 직원과의 협업 체계도 개선되며 업무 효율이 전반적으로 향상됨.

2.5 영어 커뮤니케이션

한국팀이 인도 개발자와 협업을 할 때 가장 어려운 점으로 꼽는 점이 영어 커뮤니케이션입니다. 영어라는 언어자체에 대해 익숙하지 못한 것도 있지만, 미국식 영어에 친숙해진 한국 사람에게는 인도 발음 및 억양이 낯설기 때문에, 이해하기 더 힘들 수도 있습니다. 따라서 영어 커뮤니케이션에서 발생할 수 있는 어려움을 극복하기 위해 번역 및 통역 툴을 효과적으로 활용하는 것은 한국 개발자와 인도 개발자 간의 원활한 협업을 위해 매우 중요합니다.

2.5.1 번역 툴

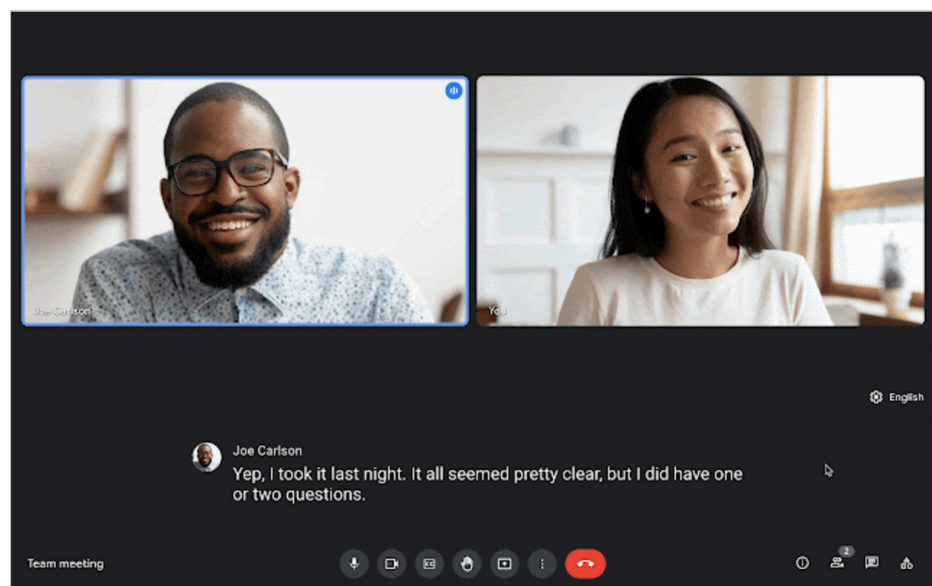
사용	툴	설명
문서 번역	Google Translate	간단한 문서나 텍스트를 빠르게 번역할 때 유용하며, 다양한 입력 방식을 지원
	DeepL	전문 문서나 기술 문서를 번역할 때 정확하고 자연스러운 결과를 제공
	ChatGPT	복잡한 문서의 번역 및 요약, 기술적 질문 해석에 유용하며, 번역 후 검토 및 수정 가능
실시간 음성 통역	Microsoft Translator	실시간 화상 회의나 프리젠테이션에서 음성 통역을 제공하여 즉각적인 소통을 지원
	Zoom 실시간 번역 기능	Zoom 회의 중 실시간 자막 및 번역을 제공하여 다국어 팀원 간의 원활한 소통 지원
메시지 번역	Slack 번역 봇	Slack 내에서 실시간으로 메시지를 번역
	Microsoft Teams 번역 기능	Microsoft Teams 내에서 실시간으로 메시지를 번역하여 다국어 팀원 간의 효율적인 소통 지원
	ChatGPT	일정적인 대화나 복잡한 질문에 대한 번역 및 해석을 지원
이메일 작성	Google Translate	간단한 이메일 작성 시 빠르게 번역하여 사용
	DeepL Translator	전문적이고 자연스러운 이메일 작성 시 유용하고, 문맥을 고려한 번역 제공
	ChatGPT	복잡한 이메일 작성 시 문맥을 고려한 자연스러운 번역과 맞춤형 응답 제공



<Google Translate>

2.5.2 번역 툴 추천 조합 및 활용 예시

사용 방안	도구 조합	활용 방법
문서 협업 시	Google Docs + DeepL Translator + ChatGPT	팀원이 Google Docs에 문서를 작성하고, DeepL 또는 ChatGPT를 통해 실시간으로 번 역을 확인하며 수정.
실시간 회의 시	Zoom + Microsoft Translator	Zoom 회의 중 Interprefy를 통해 전문 통역사 를 활용하거나, Microsoft Translator의 실시간 번역 기능을 활성화하여 다국어 소통
일상적인 소통 시	Slack + Slack 번역 봇 + ChatGPT	Slack에서 메시지를 주고받을 때 번역 봇을 통 해 실시간으로 번역하고, ChatGPT를 통해 복 잡한 질문이나 설명을 보완
이메일 작성 시	Google Translate + DeepL Translator + ChatGPT	이메일 초안을 Google Translate로 번역한 후, DeepL 또는 ChatGPT를 사용하여 자연스럽게 전문적인 문장으로 수정



<Google Meet - Live translated captions>

사례 A 사 - IT 번역 도구 도입을 통한 언어 장벽 해소

■ 문제 상황

국내 회사 A의 팀원들은 영어에 자신이 없어 문서를 번역하는 데 어려움을 겪었으며, 번역 작업이 비효율적으로 이루어져 업무 생산성이 저하됨.

■ 문제 원인

회사는 무료 IT 번역 도구를 효과적으로 활용하지 못하고 있었으며, 네이버 파

파고와 구글 번역기를 사용해 단어, 문장 단위로 수작업 번역을 반복하고 있었음.

▪ 대처 방안

• 실시간 번역 도구 도입:

- Slack 실시간 번역 Plug-in 설치: 팀원들이 한국어로 입력하면 영어가 함께 병기되고, 영어로 입력하면 한국어가 자동으로 병기되도록 설정함. (설치 시간: 3분).

• 자동 번역 확장 프로그램 적용:

- 구글 번역 Chrome Extension 설치:

· 크롬 브라우저에 구글 번역 확장 프로그램을 설치하여 웹페이지 상의 영문 텍스트를 자동으로 한국어로 번역하도록 설정.(설치 시간: 3분)

· Figma, Jira, Confluence, GitHub 등 협업 도구에서도 번역된 언어로 화면이 출력되도록 지원함.

• 동시 통번역 기능 활용:

- Google Meet 및 MS Teams 동시 통번역 기능: 실시간 회의 시 통번역 기능을 활용해 회의 중 언어 장벽을 해소함.

- Gemini 유료 서비스: Google의 Gemini를 이용해 화상회의 중 모든 음성을 번역된 자막으로 제공함.

▪ 결과

언어 장벽이 해소되면서 팀원들이 글로벌 협업에 자신감을 가지게 되었고, 번역 작업 시간이 단축되어 업무 생산성이 크게 향상되었음.

2.6 인도 개발자의 커뮤니케이션 방식 차이

인도 개발자와 한국 개발자 간의 문화적 차이로 인해 비효율적인 커뮤니케이션으로 이어질 수 있습니다. 이러한 차이는 주로 의사소통 방식, 시간 관리, 피드백 제공 방식 등에서 나타나고, 이는 프로젝트의 효율성과 팀워크에 영향을 미칠 수 있습니다.

2.6.1 인도 개발자의 커뮤니케이션 특성

■ 뛰어난 언변

- 문제 상황: 인도 개발자들의 경우, 언변이 뛰어나고 협상력이 뛰어납니다. 그러다 보니 실질적인 결과 보다는 본질에 벗어나는 이야기를 하다가 커뮤니케이션이 끝나는 경우가 발생합니다.
- 개선 방안: 인도 개발자와 업무를 할 때는 명확한 목표와 업무를 주고, 그에 대한 측정 가능한 결과를 명시하고, 결과를 중심으로 이야기하는 것이 중요합니다.

■ 지나친 긍정 답변

- 문제 상황: 인도 개발자들의 경우 'No'라는 말을 잘하지 않습니다. 문제가 명확히 파악되지 않은 상황에서도 할 수 있다라고 하는 경우가 많이 있습니다. 이에 따라 한국팀의 경우 업무를 전달한 후에 중간 관리를 하지 않고, 마지막에만 결과를 확인할 때가 많습니다. 프로젝트 마지막에 결과를 확인할 경우 예상된 결과와 상이한 경우가 발생할 수 있습니다.
- 개선 방안: 프로젝트를 전달할 경우 명확하게 전달하고, 개발자가 이해한 부분을 한번더 확인해서, 의도한 바를 정확히 이해했는지 파악하는게 중요합니다.

■ 마감 기한 미준수

- 문제 상황: 인도 개발자와 업무를 했을 때 “곧 끝납니다”라고 하더라도, 마감 기한을 지키지 못하는 상황에서 늦게 통보하여 업무에 차질을 경우가 있습니다.
- 개선 방안: 업무 범위와 일정에 대해서 이야기 할 때는 명확하게, 현재 진척도가 얼마이고, 언제까지 마칠 수 있고, 지연이 된다면 이유를 명확하게 제시하도록 하는게 중요합니다.

인도의 독특한 긍정 표현

인도 사람들과 처음 만나서 대화해 보면 인도 사람들이 고개를 좌우로 흔드는 특이한 고갯짓을 사용하는 것을 관찰 할 수 있습니다. 처음에는 개인적인 버릇인가 라고 생각했다가, 실제로 인도에 방문해 보면 모든 사람들이 고개를 좌우로 흔드는 것을 발견할 수 있습니다.

한국에서는 고개를 좌우로 흔드는 것이, 부정의 표시나, 잘 모른다는 표현으로 쓰이지만, 인도에서는 상대방의 말을 “이해했다” 또는 “동의한다”의 뜻을 나타내는 몸짓이라고 생각하시면 됩니다.

2.6.2 한국 개발자와 인도개발자의 커뮤니케이션 차이

커뮤니케이션 요소	한국인 개발자	인도인 개발자
의사소통 방식	간접적이고 조화로운 의사소통을 선호하며, 상대방의 기분을 고려함	즉시 직접적이고 개방적인 의사소통을 선호하며, 명확하게 표현~ 1시간 이내
권위와 위계 질서	위계 질서를 중시하며, 상사의 지시를 존중하고 따름	권위에 대해 도전적인 수 있으며, 상사에게도 자유롭게 의견 표현
시간 관리	시간엄수를 중요시하며, 약속 시간과 마감 기한을 철저히 지킴	시간에 대한 유연성이 높아, 약속 시간이나 마감 기한에 여유를 둠
피드백 제공 방식	완곡한 표현을 사용하여 피드백을 전달, 갈등을 피하려 함	직접적이고 솔직하게 피드백을 제공, 문제 해결을 우선시 함
회의 및 토론 스타일	회의에서 조화를 중시하며, 의견 충돌을 최소화 하려 함	활발한 토론과 아이디어 교환을 선호하며, 의견 충돌을 통한 개선을 추구
의사결정 과정	상위 권한자의 결정을 존중하고 따르는 경향이 강함	팀원 간의 논의를 통해 다양한 의견을 수렴하여 결정을 내림
비언어적 커뮤니케이션	비언어적 신호와 맥락을 중요시함. (눈치가 중요함)	비언어적 요소보다 명확한 언어적 표현을 선호함
갈등 해결 방식	갈등을 피하고 조화를 유지 하려고 함	갈등을 직접적으로 해결하려고 하며, 문제를 신속하게 처리함
협업 스타일	팀워크와 조화를 중시하며, 개인보다는 집단의 목표를 우선함	개별의 기여와 창의성을 중시하며, 팀 내 다양한 아이디어를 존중함

사례 - 객관적 성과 지표 도입을 통한 개발팀 생산성 향상

■ 문제 상황

국내 기업 A는 해외 개발자를 고용했지만, 개발자의 업무 성과를 측정할 수 있는 기준이 없어 어려움을 겪고 있었음. 국내 직원 B는 해외 개발자 C의 성

과가 낮다고 비난했으나, 회사 차원에서 이를 평가할 수 있는 명확한 수단이 없어 심증적으로만 판단할 수밖에 없는 상황이었음.

■ 문제 원인

- 개발팀은 비즈니스 밸류(사업 가치), 개발 공수 산정, 과제 배분 등의 객관적 지표 없이 주먹구구식으로 업무를 수행하고 있었음.
- 명확한 목표와 성과 지표의 부재로 개발팀이 회사의 사업적 우선순위와 무관한 작업에 몰두하게 되었음.

■ 대처 방안

1. 프로젝트의 사업 가치 명확화:

- 개발팀 프로젝트가 회사 사업에서 어떤 금전적 가치를 가지는지 사전에 정의함.
- 금전적 가치가 높은 프로젝트에 우선순위를 두고, 성공적으로 수행한 개발팀에 더 높은 성과를 부여함.
- 이를 통해 개발팀이 본인의 작업이 회사에 미치는 영향을 명확히 이해하도록 함.

2. 개발 공수 산정 및 합의:

- 다양한 이해관계자의 의견을 수렴해 **개발 기간(공수)**을 현실적으로 산정하고 합의함.
- 개발팀은 합의된 개발 기간 내에 업무를 완수하여 본인의 역량을 객관적으로 증명하도록 함.

3. 성과 지향적 문화 조성:

- 발표 역량이나 주관적 평가 대신 금전적 사업 지표를 기준으로 성과를 평가함.
- 불신을 해소하고 팀원 간 협업을 강화하면서 성과 지향적 문화를 도입함.

■ 결과

주관적 느낌이나 심증에 의존하던 평가 체계에서 벗어나, 객관적인 사업 가치와 성과 지표로 개발자 업무를 평가함으로써 팀원 간 불신이 해소됨. 개발팀은 회사의 우선순위에 맞는 프로젝트에 집중하게 되었으며, 관계 중심에서 성과 중심의 문화로 전환되어 팀 생산성이 크게 향상됨.

2.7 보안 관점에서 고려해야 할 점

인도 개발자와의 원격으로 협업할 때는 보안 관점에서 물리적 보안, 사이버 보안 그리고 인적 보안의 세가지 주요 측면을 철저히 준비를 해야 합니다. 이러한 보안 요소들은 원격 근무 환경에 국한되지 않고, 사무실에서 근무하는 국내 개발자들에게도 동일하게 적용되므로, 회사의 전반적인 보안정책에 포함되어야 합니다.

2.7.1 보안 관점

보안 계층	설명
물리적 보안	물리적 보안은 하드웨어, 장비, 사무 공간 등 물리적인 자산을 보호하는 것을 의미 합니다. 원격 근무시, 개발자가 사용하는 개인 장비나 원격 장소의 보안을 유지하는 것이 중요합니다.
사이버 보안	사이버 보안은 네트워크, 시스템, 데이터 등을 외부의 위협으로부터 보호하는 것을 의미합니다. 원격 근무 환경에서는 특히, 해킹, 데이터 유출, 악성 소프트웨어 등의 위협에 대비해야 합니다.
인적 보안	시인적 보안은 직원의 행동과 의식을 통해 보안을 유지하는 것을 의미합니다. 이는 내부자의 실수나 악의적인 행동으로부터 보안을 강화하는 데 중점을 둡니다.

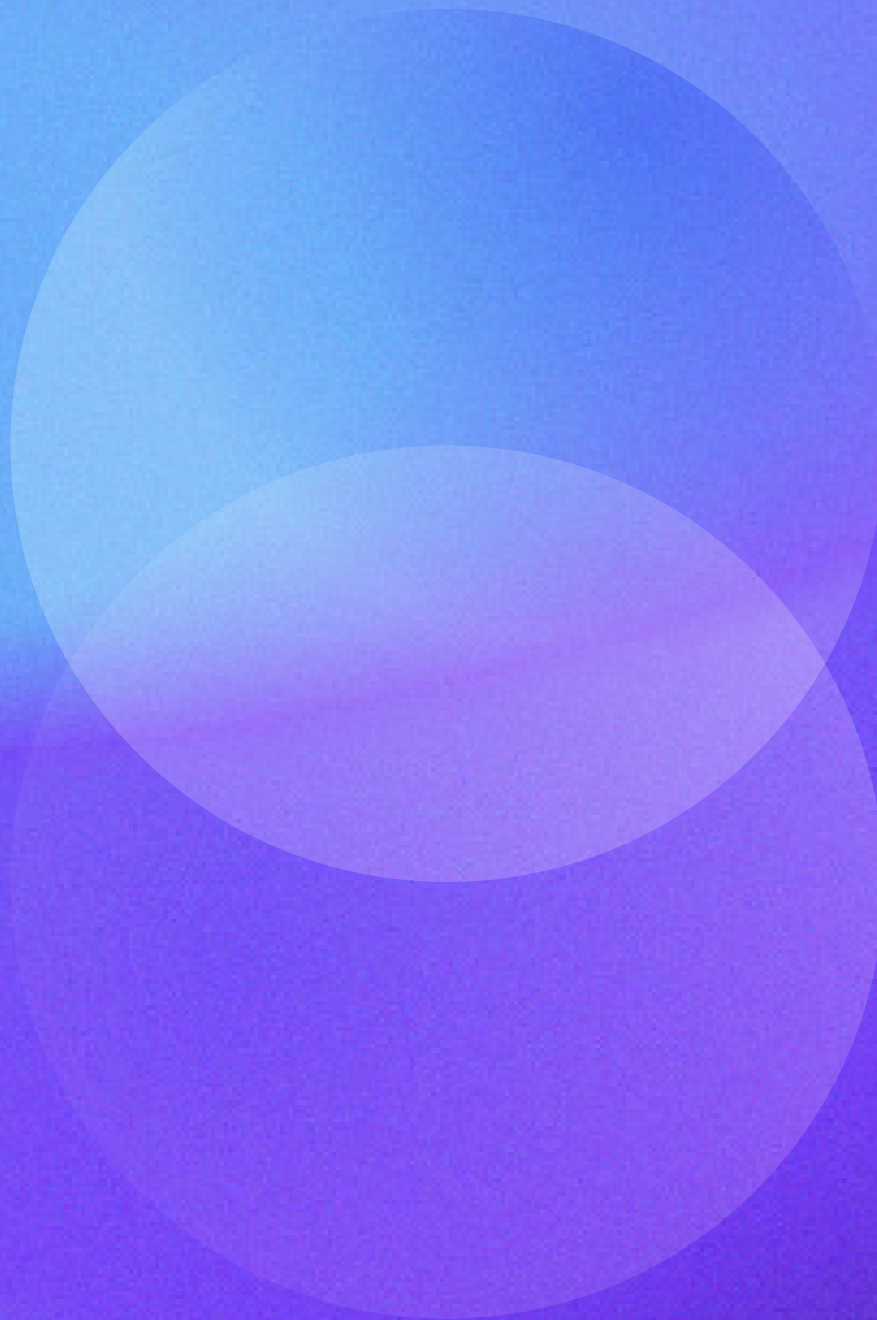
2.7.2 보안 가이드라인

보안 계층	가이드라인
물리적 보안	<ul style="list-style-type: none">- 장비 관리: 개발자가 사용하는 노트북, 스마트폰 등 장비에 대한 관리 방안을 마련하고, 분실 시 즉시 보고할 수 있는 절차를 설정합니다.- 장비 암호화: 모든 장비에 디스크 암호화 및 비밀번호 보호를 적용하여 무단 접근을 방지합니다.- 백업 장비 제공: 주요 장비의 고장이나 분실 시 신속하게 대체할 수 있도록 클라우드 등 백업 조치를 합니다.

사이버 보안	<ul style="list-style-type: none"> - VPN 사용: 모든 원격 접속은 안전한 VPN을 통해 이루어지도록 하게 하고, VPN 접속 시 다중 인증을 적용 합니다. - 강력한 비밀번호 정책: 모든 계정에 대해 강력한 비밀번호를 사용하고, 정기적으로 변경하도록 구성합니다. - 정기적인 소프트웨어 업데이트: 운영체제, 애플리케이션, 보안 소프트웨어와 방화벽을 설치하고, 정기적으로 업데이트합니다. - 데이터 암호화: 중요한 데이터는 전송 및 저장 시 암호화하여 보호합니다.
인적 보안	<ul style="list-style-type: none"> - 보안 인식 교육: 피싱, 소셜 엔지니어링 등 인적 보안 위협에 대한 교육을 정기적으로 실시합니다. - 접근 권한 관리: 최소 권한 원칙을 적용하여 필요하지 않은 데이터나 시스템에 대한 접근을 제한합니다. - 비밀 유지 계약서: 해외 개발자와의 계약 시 비밀 유지 계약을 체결하여 민감한 정보의 유출을 방지합니다. - 직원 배경 조사: 해외 개발자를 채용할 때 기본적인 배경 조사를 실시하여 신뢰할 수 있는 인재를 선발합니다.

PART 4

글로벌 애자일 팀 운영



인도 개발자와 원격으로 협업할 때 여러가지 어려운 점들이 발생할 수 있습니다. 먼저, 커뮤니케이션의 비효율성이 나타날 수 있습니다. 팀원들이 물리적으로 떨어져 있기 때문에 정기적인 소통과 정보 공유가 원활하지 않으면 오해나 정보의 누락이 발생할 가능성이 높아집니다. 또한, 프로젝트 관리 역시 어려울 수 있습니다. 이미 정해진 계획과 새로운 변경에 대한 유연성이 부족하면 예기치 않은 요구 사항이나 문제 발생 시 신속하게 대응하기 어렵습니다. 이로 인해 프로젝트 일정 지연이나 품질 저하가 발생할 수 있으며, 팀원들의 동기 부여 저하와 스트레스 증가로 이어질 수 있습니다. 또한, 피드백의 부족으로 인해 문제를 조기에 발견하고 개선하는 과정이 늦어져 전체적인 프로젝트의 효율성이 떨어질 수 있습니다.

이러한 문제점을 애자일 방식을 도입함으로써 효과적으로 해결할 수 있습니다. 애자일은 유연한 대응과 지속적인 커뮤니케이션을 강조하여 팀원 간의 소통을 원활하게 하고, 변화하는 요구사항에 빠르게 적응할 수 있도록 도와줍니다. 짧은 스프린트 단위로 작업을 나누어 진행함으로써 프로젝트의 진행 상황을 지속적으로 점검하고, 빠른 피드백을 통해 문제를 조기에 발견하고 수정할 수 있습니다. 투명한 정보 공유와 명확한 역할 분담을 통해 프로젝트 관리의 효율성을 높이고, 전체적인 프로젝트 품질을 향상시킬 수 있습니다. 따라서 원격으로 인도 개발자와 협업할 때 애자일 방식을 채택함으로써 프로젝트의 성공가능성을 크게 높일 수 있습니다.

애자일 방식에 대해 설명하기 전에 기존에 많이 진행되었던 워터폴 방식과 비교해 보면 아래와 같습니다.

구분	애자일 방법론	워터폴 방법론
개발 방식	반복적이고 점진적인 개발 방식	단계별로 순차적으로 진행되는 개발 방식
계획과 유연성	유연하게 계획을 수정 가능, 변화에 빠르게 대응	초기 계획에 엄격하게 따르며 변경에 어려움

프로젝트 단계	짧은 스프린트 단위로 여러 번 반복됨	요구사항 분석, 설계, 구현, 테스트, 유지보수의 고정된 단계
변경 관리	변경을 수용하고 스프린트마다 요구사항 조정 가능	변경이 어렵고 비용이 많이 들며, 주로 초기 단계에서 결정됨
피드백 주기	짧은 주기로 지속적인 피드백과 개선	피드백은 주로 프로젝트 완료 후에 이루어짐
팀 협업	팀원 간의 긴밀한 협업과 자율성 강조	각 단계별로 역할이 분리되어 있고, 상하 관계가 명확함
고객 참여	고객과의 지속적인 소통과 참여를 통해 요구사항 반영	고객의 참여가 주로 초기 요구사항 정의 단계에 한정됨
문서화	필요한 최소한의 문서화, 유연한 문서 관리	상세한 문서화가 필수적, 각 단계별로 철저한 문서 작성 필요
리스크 관리	초기부터 지속적으로 리스크를 관리하고, 문제를 빠르게 해결	리스크가 주로 프로젝트 후반부에 발견되어 해결이 어려움
적합한 프로젝트 유형	변화가 많고, 요구사항이 불확실한 프로젝트	요구사항이 명확하고 변경 가능성이 낮은 프로젝트

01

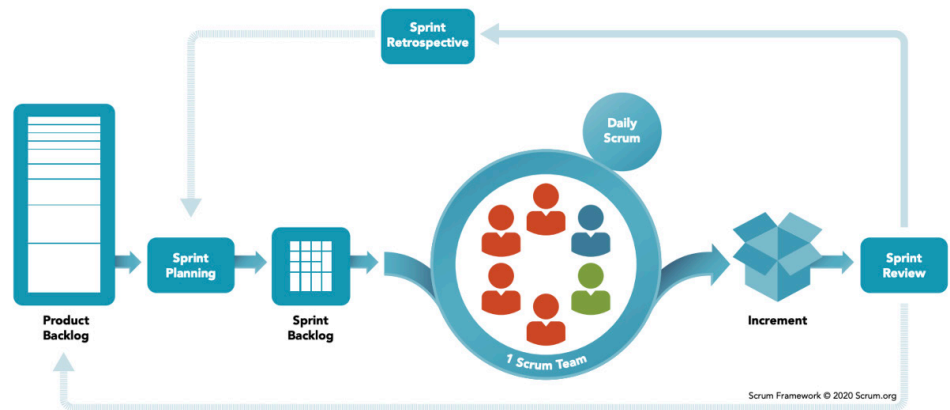
애자일 방법론

애자일 방법론은 소프트웨어 개발에서 유연성과 협업을 중시하는 접근 방식으로, 짧은 주기인 스프린트를 통해 지속적으로 기능을 개선하고 사용자 피드백을 반영합니다. 이를 통해 변화하는 요구사항에 신속하게 대응하고, 팀간의 효율적인 커뮤니케이션을 촉진하며, 높은 품질을 지속적으로 제공하는 것을 목표로 합니다. 애자일에서는 고객 만족을 최우선으로 하며, 자율적인 팀워크와 지속적인 개선을 강조합니다.

스크럼 방법론의 활용

스크럼은 애자일에서 가장 많이 도입한 방법론으로, 정해진 짧은 주기의 스프린트를 통해 팀이 목표를 설정하고, 지속적인 피드백과 협업을 바탕으로 점진적으로 제품을 개발하며, 정기적인 회의를 통해 진행 상황을 점검하고 개선해 나가는 프레임워크입니다.

스크럼은 명확하게 정의된 3개의 역할, 5가지의 정기적인 이벤트, 그리고 3가지 핵심 산출물을 통해 팀이 효율적으로 협업할 수 있습니다.



<스크럼 프레임워크>

1.1 스크럼의 3가지 역할

스크럼 마스터(Scrum Master)

스크럼 마스터는 팀의 코치 같은 역할을 합니다. 팀이 스크럼 방식을 잘 다룰 수 있도록 도와주고, 팀이 겪는 문제를 해결해줍니다. 예를 들어, 팀 회의가 원활하게 진행되도록 조율하고, 팀이 방해받지 않도록 외부의 방해 요소를 제거해줍니다. 스크럼 마스터는 팀이 더 효율적으로 일할 수 있도록 지원합니다.

1.1.1 제품 책임자(Product Owner)

제품 책임자는 프로젝트의 비전과 목표를 설정하고 관리하는 사람입니다. 쉽게 말해, 무엇을 만들어야 하는지, 어떤 기능이 가장 중요한지를 결정하는 역할을 합니다. 예를 들어, 새로운 스마트폰 앱을 개발한다고 하면, 제품 책임자는 어떤 기능이 들어가야 할지, 사용자들이 무엇을 원하는지 정리하고 우선순위를 매깁니다. 제품 책임자는 팀과 이해관계자(예: 고객, 경영진) 간의 소통을 담당 합니다.

1.1.2 개발팀(Development Team)

개발 팀은 실제로 개발 작업에 참여/수행하는 사람들입니다. 이 팀에는 개발자만 포함되었는 것이 아니라 디자이너, 개발자, 테스터 등 다양한 역할의 사람들로 구성이 될 수 있습니다. 개발팀은 제품 책임자가 정한 목표를 바탕으로 실제 제품을 만듭니다. 예를 들어, 스마트폰 앱을 개발할 때, 개발 팀은 앱의 기능을 디자인하고, 프로그래밍하고, 테스트 등의 작업을 합니다.

많은 한국 회사의 경우 스크럼 방식을 도입했다고 하지만, 제품 관리자와 스크럼 마스터가 따로 없는 경우가 많습니다. 이러한 경우 개발자들이 요구사항을 정리하고, 우선순위를 정하는 경우가 많고, 많은 경우 비즈니스의 우선순위, 즉 고객의 우선순위에 근거하여 개발을 진행하지 못하는 경우가 많습니다.

1.2 스크럼의 5가지 이벤트

스크럼에는 5가지 주요 이벤트가 있습니다. 이 이벤트들을 통해 팀이 정기적으로 소통하고, 진행 상황을 점검하며, 지속적으로 제품 및 프로세스를 개선합니다.

1.2.1 스프린트 (Sprint)

스프린트는 보통 2주에서 4주 정도의 짧은 기간으로 설정됩니다. 이 기간 동안 팀은 특정 목표를 달성하기 위해 필요한 작업을 계획하고 실행합니다. 예를 들어, 한달 동안 진행되는 스프린트에서는 새로운 기능을 개발하거나 문제를 해결하는데 집중할 수 있습니다. 스프린트가 끝나면 팀은 그 동안의 성과를 검토하고 다음 스프린트를 준비합니다. 스프린트는 쉽게 생각하면, 짧은 기간 동안 집중적으로 작업하는 시간 블록이라고 생각하시면 됩니다.

1.2.2 스프린트 계획 회의 (Sprint Planning)

스프린트 계획 회의는 다음 스프린트 동안 무엇을 할지 결정하는 회의입니다. 스프린트가 시작되기 전에 이번 스프린트에서 달성할 목표와 필요한 작업을 논의하고 결정합니다. 팀원들은 어떤 작업을 우선순위에 두고, 각 작업을 완료하기 위해 얼마나 시간이 필요한지 논의 및 추정합니다. 이를 통해 스프린트 동안 집중해서 작업할 업무가 정해집니다.

1.2.3 데일리 스크럼 (Daily Scrum)

데일리 스크럼은 매일 짧게 진행되는 팀 회의로, 현재 진행 상황을 공유하는 자리입니다. 보통 15분 정도로 시간을 정하고, 매일 같은 시간에 모여서 미팅을 진행합니다. 각 팀원들은 자신이 어제 어떤 작업을 했는지, 오늘 무엇을 할 계획인지, 그리고 현재 업무 진행하는데 있어 방해 요소가 있는지 공유합니다. 이를 통해 팀원들의 진행 상황을 파악하고, 현재 프로젝트에 방해되는 이슈들을 신속하게 파악하고 해결해 나갈 수 있습니다.

1.2.4 스프린트 리뷰 (Sprint Review)

스프린트 리뷰는 스프린트가 끝난 후, 완성된 작업을 검토하고 피드백을 받는 시간입니다. 이 회의에서는 스프린트 동안 작업 완료된 작업, 즉 산출물을 이해관계자 또는 고객에게 시연하고, 피드백을 받습니다. 피드백을 통해 제품의 개선점이나 추가 요구사항을 파악할 수 있으며, 이는 다음 스프린트의 계획에 반영할 수 있습니다. 스프린트 리뷰는 제품이 올바른 방향으로 나아가고 있는지 확인하는 중요한 단계입니다.

1.2.5 스프린트 회고 (Sprint Retrospective)

스프린트 회고는 팀이 지난 스프린트를 돌아보고, 팀간의 협업 방식에서 잘된점과 개선할 점을 찾는 시간입니다. 이 회의에서는 제품과 서비스에 대해서 이야기하지 않고, 팀간의 커뮤니케이션 또는 프로세스에 대해서 어떤 부분이 효과적이었고, 어떤 문제가 있었는지, 앞으로 어떻게 더 나아질 수 있을지를 이야기 합니다. 이를 통해 지속적으로 프로세스를 개선하고, 더 효율적으로 일할 수 있는 환경을 만들어 갑니다.

1.3 스크럼의 3가지 산출물

스크럼에서 프로젝트의 진행 상황과 작업을 체계적으로 관리하기 위해 세 가지 주요 산출물을 관리합니다.

1.3.1 제품 백로그 (Product Backlog)

제품 백로그는 프로젝트에서의 필요한 모든 작업과 기능, 즉 요구사항 목록입니다. 쉽게 말해서 프로젝트의 “할 일 목록”이라고 생각하면 쉽습니다. 이 목록에는 제품에 추가되어야 할 모든 기능 부터 수정 사항, 개선점 등이 포함됩니다. 예를 들어, 스마트폰 앱을 개발할 때, 제품 백로그에는 사용자 로그인 기능, 채팅 기능, 알림 설정 등 다양한 항목이 나열될 수 있습니다. 제품 책임자가 이 목록을 관리하며, 각 항목의 우선순위를 정해 가장 중요한 작업부터 차례로 진행할 수 있도록 합니다. 제품 백로그는 지속적으로 업데이트 되며, 프로젝트가 진행됨에 따라 새로운 요구사항이 추가되거나, 기존 항목의 우선순위가 변경될 수 있습니다.

1.3.2 스프린트 백로그 (Sprint Backlog)

스프린트 백로그는 바로 해야 할 작업의 세부 목록입니다. 제품 백로그에서 이번 스프린트(보통 2~4주) 동안 완료할 작업을 선택하여 만든 목록입니다. 이를 통해 팀은 짧은 기간 동안 집중적으로 작업할 목표를 설정할 수 있습니다. 예를 들어, 한 달 동안 진행되는 스프린트에서는 “사용자 로그인 기능 구현”, “채팅 기능 테스트”, “알림 설정 UI 디자인” 등의 구체적인 작업들이 스프린트 백로그에 포함될 수 있습니다. 스프린트 백로그는 팀이 스프린트 동안 어떤 작업을 언제까지 완료할 것인지를 명확히 이해하고, 작업의 진행 상황을 지속적으로 추적할 수 있게 합니다.

1.3.3 증가분 (Increment)

증가분은 스프린트 동안 완료된 모든 작업의 결과물입니다. 이는 “작동 가능한 제품의 일부분”으로, 사용자가 실제로 사용할 수 있는 상태를 기준으로 합니다. 예를 들어, 스프린트가 끝나면 사용자 로그인 기능이 구현되어 사용자가 로그인을 할 수 있게 되고, 채팅 기능이 테스트되어 실제로 메시지를 주고받을 수 있는 상태가 되면 증가분으로 포함 시킵니다. 해당 증가분은 이전에 개발된 증가분에 추가되어 점진적으로 제품을 완성해 가게 됩니다.

02

코드 리뷰 및 버전 관리

해외 개발자와 원격으로 협업할 때 코드 리뷰 및 버전 관리는 프로젝트의 성공과 팀의 효율성을 보장하는 데 필수적인 요소입니다. 서로 다른 시간대의 개발자들이 작업할 때, 코드 리뷰를 통해 일관된 코딩 표준을 유지하고, 코드의 가독성과 유지보수성을 향상시킬 수 있습니다. 원격 협업에서는 물리적인 거리로 인해 의사소통의 빈틈이 생길 수 있는데, 코드 리뷰와 버전관리를 통해 이러한 문제를 최소화하고, 팀 전체의 협업 효율성을 극대화할 수 있습니다.

2.1 코드 리뷰

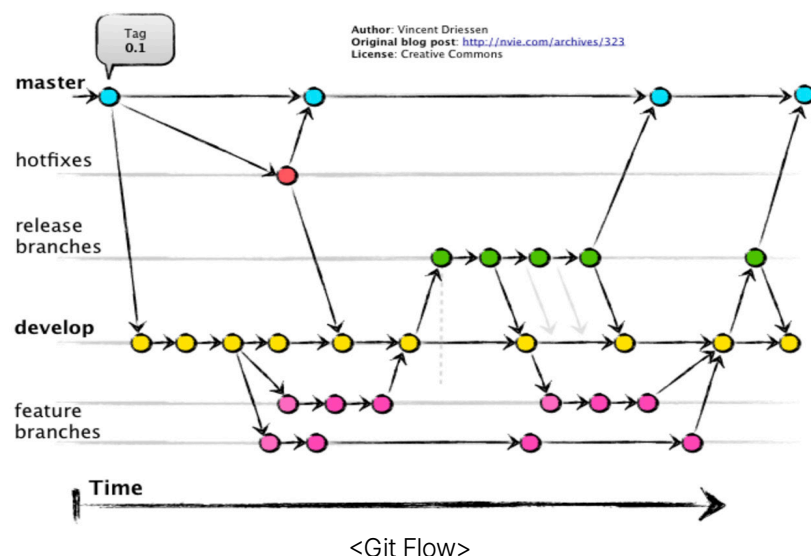
아래는 해외 개발자와 원격으로 협업할 때 효과적인 코드 리뷰 프로세스를 단계별로 정리한 것입니다. 각 단계마다 수행할 작업, 책임자, 사용 가능한 도구, 그리고 주요 고려사항을 포함하고 있습니다.

단계	설명	책임자/역할	주요 고려 사항
코드 작성	개발자가 기능을 구현하고 코드를 작성합니다.	개발자	코딩 표준 및 가이드라인 준수, 충분한 주석과 문서화
커밋 및 푸시	작성한 코드를 로컬 저장소에 커밋하고 원격 저장소로 푸시합니다.	개발자	의미 있는 커밋 메시지 작성 작은 단위의 커밋으로 변경 사항 분리
풀 리퀘스트 생성	개발자가 변경 사항을 팀에 공유하기 위해 풀 리퀘스트를 생성 합니다.	개발자	변경 사항에 대한 명확한 설명 작성 관련 이슈 번호 링크
자동화 테스트 실행	CI/CD 파이프라인을 통해 자동화된 테스트가 실행됩니다.	CI/CD 시스템	모든 테스트가 통과해야 코드 리뷰 진행 테스트 커버리지 확인
코드 리뷰 요청	리뷰어에게 코드 리뷰를 요청합니다.	개발자, 팀 리드	리뷰어 선정 시 경험과 역할 고려 리뷰 요청 시 명확한 피드백 요청
코드 리뷰 수행	리뷰어가 코드를 검토하고 피드백을 제공합니다.	코드 리뷰어	코드의 가독성, 유지보수성, 성능, 보안등 다양한 측면 검토, 긍정적인 건설적인 피드백 제공
피드백 반영 및 수정	개발자가 리뷰어의 피드백을 반영하여 코드를 수정합니다.	개발자	피드백 사항을 정확히 이해하고 반영 수정 사항에 대한 추가 리뷰 요청 가능
승인 및 머지	리뷰어가 모든 피드백이 반영되었는지 확인하고, 코드를 메인 브랜치에 머지합니다.	코드 리뷰어, 팀 리드	머지 전 최종 테스트 수행 충돌 해결 및 코드 통합 확인
배포 준비	머지된 코드를 배포 환경으로 준비합니다.	DevOps 엔지니어, 팀 리드	배포 스크립트 및 설정 확인, 롤백 계획 수립
배포 및 모니터링	코드를 실제 운영 환경에 배포하고, 모니터링을 통해 문제를 신속히 감지 및 대응합니다.	DevOps 엔지니어, 팀 리드	배포 후 기능 정상 동작 확인, 모니터링 도구를 통한 실시간 상태 확인 및 알림 설정

2.2 버전 관리

버전 관리는 프로젝트의 일관성, 코드 품질, 그리고 팀 간의 효율적인 협업을 보장하는데 필수적인 요소입니다. 버전 관리를 통해 여러가지 개발자가 동일한 코드베이스에서 작업할 수 있으며, 변경 사항을 체계적으로 추적하고 관리할 수 있습니다. 아래 표는 Gitflow 브랜칭 모델의 각 브랜치 유형에 대한 설명과 목적을 정리한 것입니다.

브랜치 유형	설명	목적
Main	프로젝트의 안정적인 배포 가능한 상태를 유지하는 기본 브랜치입니다.	항상 프로덕션 환경에 배포 가능한 코드를 유지합니다.
Developer	개발 중인 모든 기능이 통합되는 브랜치로, 다음 배포를 위한 준비가 되는 브랜치입니다.	새로운 기능이 개발되고 통합되는 중심 브랜치로, 안정적인 배포 전 최종 테스트를 거칩니다.
Feature	개별 기능을 개발하기 위한 브랜치로, developer 브랜치에서 분기하여 사용 됩니다.	특정 기능이나 작업을 독립적으로 개발하고, 완료 후 develop 브랜치로 머지 합니다.
Release	다음 릴리즈 준비를 위한 브랜치로, develop 브랜치에서 분기하여 사용 됩니다.	최종 테스트, 버그 수정, 릴리즈 노트 작성 등을 수행하여 프로덕션 배포를 준비합니다.
Hotfix	프로덕션에서 발견된 긴급 버그를 수정하기 위한 브랜치로, main 브랜치에서 직접 분기하여 사용됩니다.	긴급한 버그를 신속하게 수정하고, 수정된 코드를 main과 develop 브랜치에 반영합니다.



사례

■ 문제 상황

B 개발팀은 코드 리뷰 과정에서 의견 충돌이 잦고, 코드 품질이 일정하지 않아 유지보수에 어려움을 겪음. 코드 리뷰 참여율이 저조하고, 리뷰 기간이 평균 3일 이상 소요되었습니다.

■ 개선 방안

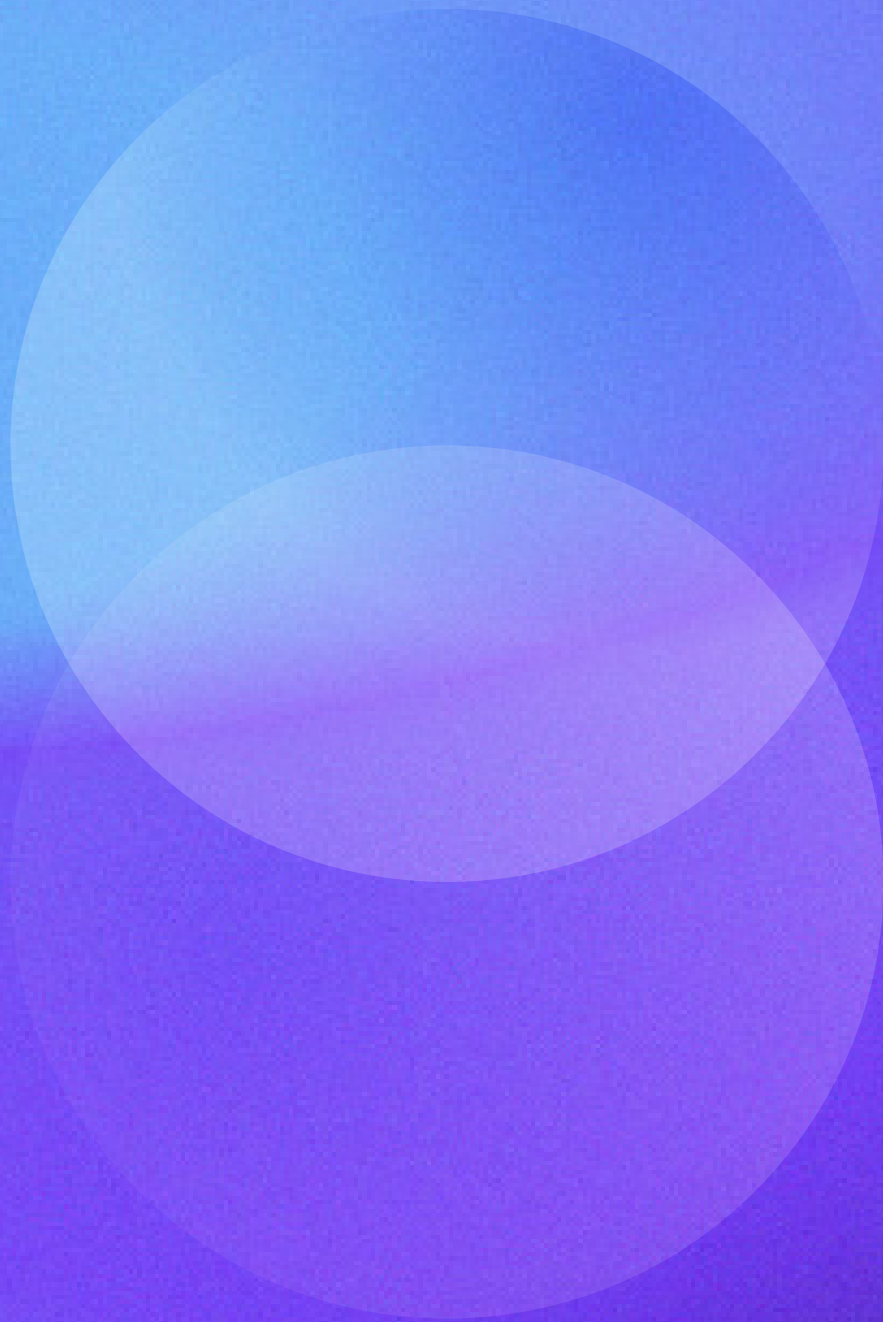
- 코드 리뷰 가이드라인 명확화: 코드 리뷰 시 중점적으로 확인해야 할 사항 (ex. 코드 스타일, 네이밍 컨벤션, 성능, 보안)에 대한 가이드라인을 명확히 하여 리뷰 기준을 통일했습니다.
- 코드 리뷰 도구 도입: GitHub의 코드 리뷰 도구를 도입하여 리뷰 과정을 체계화하고, 리뷰 히스토리를 관리했습니다.
- 코드 작성 컨벤션 통일: Google Java Style Guide, Python PEP 8과 같은 코드 작성 컨벤션을 도입하여 코드 스타일을 통일하고, 코드 품질을 향상시켰습니다.
- 코드 품질 교육: 코드 작성 컨벤션, 코드 리뷰 가이드라인, 좋은 코드 작성법 등에 대한 교육을 통해 개발팀 전체의 역량을 향상시켰습니다.

■ 결과

코드 리뷰 참여율이 90% 이상으로 증가하고, 리뷰 기간이 평균 1일 이내로 단축됨. 코드 품질이 향상되어 유지보수가 용이해졌습니다.

PART 5

성공적인 원격 근무를 위한 모범 사례



- ☐ Dell
- ☐ Automattic
- ☐ Zapier
- ☐ Doist
- ☐ GitLab

Dell

Dell: “Connected Workplace” 프로그램 운영

▪ 회사 소개

Dell Technologies는 재택근무를 회사 문화와 성장 전략의 중요한 부분으로 받아들였으며, 이를 위해 인프라를 구축해 생산성과 직원 만족도를 높였습니다. EMC와의 합병을 통해 세계 최대의 민간 기술 기업으로 성장한 Dell은 2020년 ‘Connected Workplace Program 2.0’을 도입해 재택근무 환경을 개선했습니다.

▪ 협업 방법

• 강력한 리더십 지원

- 경영진이 유연하게 근무하며 원격 근무를 적극 지지합니다.

• 직원 신뢰

- 직원들이 업무와 우선순위를 관리할 수 있도록 신뢰를 부여합니다.

• 결과 중심

- 전통적인 생산성 지표보다 결과를 중시하는 문화 구축

Automattic: 글로벌 분산 팀 운영**▪ 회사 소개**

Automattic은 워드프레스(Woocommerce)의 모회사로, 전 세계에 분산된 팀원들과 원격 근무를 통해 운영되는 회사입니다. Automattic은 92개 이상의 국가에서 팀원들이 원격으로 일하고 있으며, 24시간 내내 활동중입니다.

▪ 협업 방법**• 유연한 작업 시간과 비동기식 커뮤니케이션**

- 유연 근무제: 직원들이 자신의 시간대와 생활 방식에 맞게 일할 수 있도록 유연 근무제를 운영합니다. 이는 다양한 시간대에 걸친 협업을 가능하게 합니다.

- 비동기식 커뮤니케이션: Slack, P2(자체 개발한 내부 블로그 도구), 이메일 등을 사용하여 비동기식 커뮤니케이션을 장려합니다. 이를 통해 시간대 차이로 인한 문제를 최소화합니다.

• 정기적인 만남과 팀 빌딩

- 연간 워드캠프(WordCamp): 매년 전 직원이 한자리에 모이는 워드캠프를 개최하여 팀원 간의 유대감을 강화하고, 회사의 비전을 공유합니다.

- 팀 빌딩 활동: 정기적인 가상 팀 빌딩 활동과 게임을 통해 팀원 간의 유대감을 유지하고, 협업을 촉진합니다.

• 기술 도구의 활용

- WordPress 및 P2: 문서화와 프로젝트 관리를 수행하여 모든 팀원이 실시간으로 정보를 공유하고 협업할 수 있습니다.

- Zoom: 화상 회의를 통해 정기적인 팀 회의와 브레인스토밍 세션을 진행합니다.

• 언어 장벽 극복

- 명확한 문서화: 모든 중요한 정보와 절차를 명확하게 문서화하여 팀원들이 쉽게 이해할 수 있도록 합니다.

- 번역 도구 활용: 필요 시 Google Translate와 같은 번역 도구를 사용하여 언어 장벽을 극복합니다.

Zapier: 전 세계에 걸쳐 전적으로 원격 근무

▪ 회사 소개

Zapier는 40개국에 걸쳐 800명 이상의 팀원이 있는 100% 원격 근무 회사입니다. Zapier는 전 세계에 있는 직원들이 유연하게 일할 수 있도록 지원합니다.

▪ 협업 방법

- 자기 동기부여가 강한 인재 채용
 - 자기 동기부여가 강한 인재를 채용하여 자율적으로 업무를 수행하도록 장려합니다.
- 명확한 커뮤니케이션 및 문서화 강조
 - 모든 업무와 절차를 명확하게 문서화하고, 명확한 커뮤니케이션 기준을 설정합니다.
- 정기적인 가상 팀 빌딩 활동
 - 정기적인 가상 팀 빌딩 활동을 통해 팀원 간의 연결을 유지하고 협업을 촉진합니다.

Doist: Todoist 및 Twist 제작 회사

▪ 회사 소개

Doist는 Todoist 및 Twist 생산성 앱을 제작하는 회사로 전적으로 원격 팀을 운영합니다. Doist는 전 세계에 걸쳐 25개의 국가에서 분산된 팀원들과 협업합니다.

▪ 협업 방법

- 비동기적 커뮤니케이션 모델
 - 비동기적 커뮤니케이션을 통해 다양한 시간대에 있는 팀원들이 효율적으로 협업할 수 있도록 합니다.
- 워크라이프 밸런스 및 직원 복지
 - 워크라이프 밸런스와 직원 복지에 집중하여 직원들이 만족하며 일할 수 있도록 지원합니다.
- 강력한 회사 가치
 - 원격 근무와 일치하는 강력한 회사 가치를 바탕으로 운영됩니다.

GitLab: 100% 원격 근무로 글로벌 팀 운영

▪ 회사 소개

GitLab은 전 세계에 분산된 팀원들과 100% 원격 근무를 통해 운영되는 소프트웨어 개발 플랫폼입니다. GitLab은 원격 근무의 효율성을 극대화하고, 글로벌 인재를 활용하여 빠르게 성장하고 있는 회사입니다.

▪ 협업 방법

• 명확한 커뮤니케이션 규칙 설정

- 공개된 핸드북: 모든 직원이 참고할 수 있는 상세한 핸드북을 공개하여 회사의 모든 규칙과 절차를 명확히 합니다.
- 비동기식 커뮤니케이션: 시간대 차이를 극복하기 위해 비동기식 커뮤니케이션을 강조합니다.

• 협업 도구의 적극적 활용

- GitLab 자체 플랫폼: GitLab 플랫폼을 사용하여 소스 코드 관리를 포함한 모든 개발 과정을 관리합니다.
- Zoom: 정기적인 화상 회의를 위해 Zoom을 사용하여 팀원들이 중요한 회의와 브레인스토밍 세션을 진행할 수 있도록 합니다.

• 문화적 차이와 언어 장벽 극복

- 문화 인식 교육: 팀원들에게 정기적인 문화 인식 교육을 제공하여 서로의 문화를 이해하고 존중할 수 있도록 합니다.
- 실시간 번역 도구 사용: 필요할 때 실시간 번역 도구를 사용하여 언어 장벽을 극복하고, 모든 팀원이 이해할 수 있도록 합니다.

• 코드 리뷰와 지속적 통합(CI)

- 상호 코드 리뷰: 모든 코드 변경 사항은 PR(Pull Request)을 통해 제출되고, 팀원들이 상호 리뷰를 통해 코드의 품질을 보장합니다.
- 지속적 통합(CI): GitLab의 CI/CD 기능을 통해 코드가 변경될 때마다 자동으로 빌드하고 테스트하여 코드의 안정성을 유지합니다.

▪ 성과

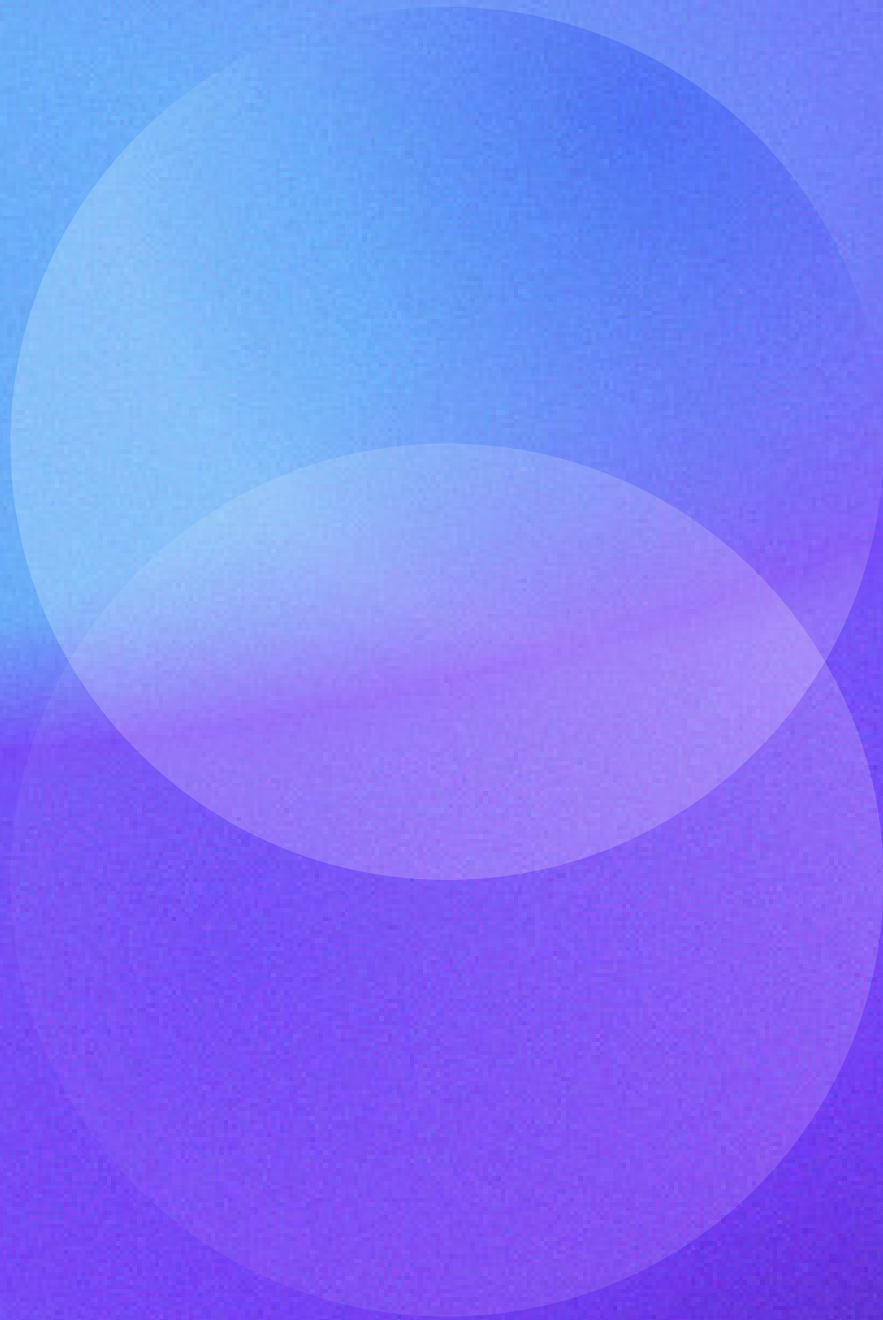
GitLab은 전 세계에 분산된 팀원들과 100% 원격 근무를 통해 높은 생산성과 협업 효율성을 유지하고 있습니다. 이를 통해 빠르게 성장하고 있으며, 글로벌 시장에서의 경쟁력을 확보하고 있습니다.

▪ 원격 근무 성공을 위한 핵심 요소

1. 직원 신뢰: 직원에 대한 신뢰를 우선시하여 최상의 결과를 도출하도록 동기 부여합니다.
2. 품질 중시: 엄격한 생산성 지표보다 높은 품질의 업무 산출물을 강조합니다.
3. 효과적인 커뮤니케이션: 명확한 커뮤니케이션 기준 및 적절한 도구 사용을 장려합니다.
4. 강한 회사 문화: 물리적 거리에도 불구하고 소속감과 공유 목적을 구축합니다.
5. 유연성: 개인의 상황과 생산성 패턴에 맞게 일할 수 있도록 허용합니다.
6. 리더십 지원: 경영진과 관리자가 원격 근무 실천을 수용하고 모범을 보입니다.
7. 결과 중심: 근무 시간이나 물리적 존재보다는 성과에 집중합니다.
8. 기술 투자: 원격 협업을 지원하는 필수 도구와 플랫폼을 제공합니다.

PART 6

인도 개발자 온보딩 체크리스트



06

인도 개발자 온보딩 체크리스트

인도 개발자를 채용하고 온보딩하는 과정은 체계적이고 철저하게 관리되어야 합니다. 여기서는 채용 전, 채용 후 첫 주, 첫 달, 3개월까지의 일간 체크리스트를 제 공합니다. 스크럼 프로젝트 진행 부분도 포함하여 구체적으로 설명하겠습니다.

01

채용 전 체크리스트

단계	체크 리스트	체크
채용 공고 작성 및 게시	구체적인 역할과 책임 명시	<input type="checkbox"/>
	필요 기술과 경험 요건 작성	<input type="checkbox"/>
	회사 소개 및 원격 근무 정책 설명	<input type="checkbox"/>
서류 검토 및 인터뷰 준비	이력서와 포트폴리오 검토	<input type="checkbox"/>
	기술 인터뷰 질문 목록 작성	<input type="checkbox"/>
	문화 적합성 인터뷰 질문 목록 작성	<input type="checkbox"/>
인터뷰 진행	1차 인터뷰: 기술 역량 평가	<input type="checkbox"/>
	2차 인터뷰: 문화 적합성 평가	<input type="checkbox"/>
	3차 인터뷰: 최종 면접 및 조건 협의	<input type="checkbox"/>
채용 결정 및 오퍼 제공	합격자에게 오퍼 레터 발송	<input type="checkbox"/>
	급여, 근무 조건, 복지 혜택 명시	<input type="checkbox"/>
채용 전 준비 사항	NDA 및 기타 서류 준비	<input type="checkbox"/>
	필요한 장비 및 소프트웨어 목록 작성	<input type="checkbox"/>
	회사 이메일 및 계정 생성 준비	<input type="checkbox"/>

02

채용 후 첫 주 체크리스트

일차	체크 리스트	체크
1일차	환영 이메일 발송	<input type="checkbox"/>
	필요한 서류 안내 (NDA, 계약서 등)	<input type="checkbox"/>
	계정 생성 및 접근 권한 부여 (이메일, 프로젝트 관리 도구 등)	<input type="checkbox"/>
	첫 주 일정 안내	<input type="checkbox"/>
	팀 소개 (화상 회의)	<input type="checkbox"/>
2일차	회사 문화 및 정책 소개 세션 (화상 회의)	<input type="checkbox"/>
	주요 도구 및 시스템 사용법 교육 (화상 회의 또는 문서 제공)	<input type="checkbox"/>
3일차	초기 프로젝트 배정 및 멘토 지정	<input type="checkbox"/>
	스크럼 방법론 교육 (스크럼 마스터 또는 팀 리더 주관)	<input type="checkbox"/>
	첫 스프린트 계획 회의 (스크럼 보드 설정, 백로그 작성)	<input type="checkbox"/>
4일차	개발 환경 설정 지원 (IDE, 버전 관리 도구 등)	<input type="checkbox"/>
	첫 데일리 스탠드업 참여	<input type="checkbox"/>
5일차	초기 작업 할당 및 진행	<input type="checkbox"/>
	주간 회의 참여 및 진행 상황 보고	<input type="checkbox"/>

03

첫 달 체크리스트

일차	체크 리스트	체크
2주차	2주차 피드백 제공 (주 1회 피드백 세션)	<input type="checkbox"/>
	진행 상황 점검 및 질문 응답	<input type="checkbox"/>
	스크럼 미팅 (스프린트 계획, 데일리 스탠드업, 스프린트 리뷰)	<input type="checkbox"/>
3주차	추가 교육 세션 (기술 스택, 코드 리뷰 등)	<input type="checkbox"/>
	팀 빌딩 활동 (가상 커피 타임 등)	<input type="checkbox"/>
	3주차 피드백 제공	<input type="checkbox"/>
4주차	첫 스프린트 리뷰 및 회고 확인	<input type="checkbox"/>
	두 번째 스프린트 계획 및 업무 할당 확인	<input type="checkbox"/>
	1개월 평가 (성과, 적응도, 추가 지원 필요 사항 논의)	<input type="checkbox"/>

(3) 3개월 체크리스트

일차	체크 리스트	체크
2개월차	정기적인 피드백 세션 (주 1회)	<input type="checkbox"/>
	스크럼 미팅 계속 진행	<input type="checkbox"/>
	기술 심화 교육 세션 (특정 기술 또는 도구에 대한 심화 교육)	<input type="checkbox"/>
	2개월 평가 (성과, 적응도, 추가 지원 필요 사항 논의)	<input type="checkbox"/>
3개월차	정기적인 피드백 세션 (주 1회)	<input type="checkbox"/>
	스크럼 미팅 계속 진행	<input type="checkbox"/>
	프로젝트 참여 확대 (책임 확대, 복잡한 작업 할당)	<input type="checkbox"/>
	최종 평가 및 정신 온보딩 완료 <ul style="list-style-type: none"> • 성과 평가 (목표 달성 여부, 기술 역량 평가) • 팀 피드백 수집 (팀원들과의 협업, 문화 적응도 평가) • 추가 개발 기회 제공 (교육 프로그램, 멘토링 기회) 	<input type="checkbox"/>

마무리 글

원격 근무 환경에서 성공적으로 일하기 위한 모든 요소들을 종합적으로 살펴보았습니다. 이 가이드북은 인도 개발자 채용의 주요 이유부터 다양한 채용 방법, 효과적인 커뮤니케이션 전략, 원격 근무의 도전과 해결 방안, 글로벌 애자일 팀 운영, 성공적인 원격 근무를 위한 모범 사례, 그리고 인도 개발자의 온보딩 체크리스트까지 전반적인 내용을 다루고 있습니다. 이제, 이 모든 요소들을 하나로 묶어 결론을 내리고자 합니다.

글로벌 인재와의 원격 근무는 현대 비즈니스 환경에서 점점 더 중요해지고 있으며, 이를 성공적으로 수행하기 위해서는 체계적인 준비와 관리가 필수적입니다. 인도 개발자의 풍부한 인재 풀과 뛰어난 기술력을 활용함으로써 기업은 경제적 효율성과 글로벌 커뮤니케이션 능력을 동시에 강화할 수 있습니다. 이를 위해서는 적절한 채용 방법과 온보딩 절차를 통해 인도 개발자와의 협업을 원활하게 진행해야 하며, 효과적인 커뮤니케이션 전략과 보안 가이드라인을 준수하여 안전하고 효율적인 작업 환경을 조성해야 합니다.

또한, 애자일 방법론을 기반으로 한 글로벌 팀 운영과 코드 리뷰 및 버전 관리의 체계적인 적용은 프로젝트의 성공과 팀의 생산성을 높이는 데 중요한 역할을 합니다. 성공적인 원격 근무를 위해서는 안정적인 인프라 구축과 팀 빌딩 활동을 통해 팀원 간의 유대감을 강화하고, 지속적인 보안 관리와 인적 보안 교육을 통해 보안 위협을 최소화해야 합니다.

이 가이드북이 한국 기업들이 인도 개발자와의 협업을 성공적으로 이끌고, 글로벌 경쟁력을 강화하는 데 실질적인 도움이 되기를 바랍니다. 원격 근무 환경에서 높은 생산성과 협업 효율성을 유지하며, 인도 개발자와의 협업을 통해 조직의 목표를 효과적으로 달성하고, 글로벌 시장에서의 입지를 더욱 공고히 할 수 있기를 기대합니다. 글로벌 인재와의 협업은 단순한 인력 구성의 변화가 아니라, 더 나은 업무 환경과 더 큰 성과를 이끌어내는 혁신의 기회입니다. 이를 통해 조직과 구성원 모두가 성장하고 발전할 수 있기를 바랍니다.

앞으로도 변화하는 글로벌 시장 속에서 유연하고 효과적인 협업 방식을 지속적으로 모색하며, 다양한 문화와 배경을 가진 인재들과 함께 성공적인 프로젝트를 완수해 나가시길 응원합니다. 본 가이드북이 여러분의 여정에 든든한 동반자가 되기를 바랍니다.

감사합니다.

