

2026-1학기 25기 심비우스 북클럽 운영보고서

북클럽 팀명	NextDev					
일시/장소	일시	2026.04.07			장소	도서관 4층
참여 인원	학과	빅데이터	학번	2024*****	이름	박*진
	학과	컨텐츠IT	학번	2024*****	이름	이*운
	학과	스마트IoT	학번	2024*****	이름	김*비
	학과	소프트웨어 AI 계열	학번	2026*****	이름	황*민
	참석인원(명)	4 / 4		참가자 대표	박*진 (서명)	
도서	도서명	오늘부터 개발자			저자	김병욱
	출판사	천그루숲	출판연도	2021	번역자	
오늘의 북클럽 주제	비전공자도 개발자가 될 수 있는 시대					
활동 내용 & 방법						
<p>시가 코드를 대신 작성해주는 시대에 개발자라는 직업의 진짜 의미와 경쟁력이 무엇인지 탐구하는 출발점으로서, 비전공자의 시선에서 개발자가 된다는 것의 의미를 담은 이 책을 함께 읽기로 했다. 우리는 모두 개발을 배우고 있지만, 정작 '개발자란 어떤 사람인가', '어떤 방향의 개발자가 되고 싶은가'에 대해 깊이 생각해본 적이 없었기에, 이 책을 통해 각자 미처 정리하지 못했던 생각을 발견할 수 있을 것이라 기대했다.</p> <p>책을 읽은 후, 인상 깊었던 내용을 나누기 전에 먼저 각자 책 소감을 자유롭게 이야기하는 시간을 가졌다. 프론트엔드, 백엔드, 데브옵스 등 다양한 개발자 직군을 새롭게 알게 된 점, 유지보수의 중요성, 그리고 스스로 어떤 방향의 개발자가 되고 싶은지에 대한 각자의 생각을 나눴다. 이후 사전에 준비한 핵심 주제를 바탕으로 본격적인 토론을 진행했다. 주요 토론 주제는 다음과 같았다.</p> <p>첫 번째로 '개발자에게 중요한 역량은 무엇인가?' 를 논의하며, 문제를 정의하는 능력과 코드를 깔끔하게 짜는 능력 중 어느 것이 더 중요한지, 그리고 AI 시대에 코드를 읽고 설명할 수 있는 능력이 왜 차별점이 되는지 이야기를 나눴다. 두 번째로 '개발 진입 장벽 하락은 위기인가, 기회인가?' 를 주제로 전공자의 희소성과 차별성, 그리고 현재의 학습이 여전히 의미 있는지에 대한 각자의 생각을 공유했다. 세 번째로 '전공자만의 강점은 무엇인가?' 를 논의하며 공학적 원리를 이해하고 설계할 수 있는 능력에 대해 이야기했다.</p>						
활동 결과						
<p>-책 소감</p> <p>박*진 : 개발자의 종류 중에 데브옵스 개발자 직군을 새로 알게되었다.</p> <p>지금은 수업시간에 배운 것들만 집중했는데 이제는 개발자가 되기 위해서 스스로 더 공부하는 시간을 가져야겠다고 생각했다. 여러 개발자 직군 중에서도 데이터 사이언티스트가 되고 싶다. 막연하게 느껴지거나 말로 설명하기 어려운 것들이 데이터를 통해 어느 정도 정형화되고 시각화될 수 있다는 점이 굉장히 흥미롭다. 예를 들어 "요즘 사용자들이 이 기능을 불편해하는 것 같다"는 감각적인 느낌도, 데이터로 분석하면 실제로 어디서 이탈이 많이 발생하는지, 어떤 흐름에서 문제가 생기는지를 구체적으로 보여줄 수 있다. 그렇게 내가 분석한 데이터가 프로젝트의 방향을 결정하고, 결과적으로 사용자가 더 나은 경험을 하게 된다면 단순히 코드를 짜는 것과는 또 다른 종류의 보람이 있을 것 같다. 또 데이터 분석은 개발 지식뿐 아니라 통계적 사고, 도메인 이해, 스토리텔링 능력까지 요구한다는 점에서 계속 성장할 수 있는 분야라는 것도 매력적으로 느껴진다. 숫자 안에서 의미를 찾아내는 그 과정이 단순한 작업이 아니라 하나의 문제 해결</p>						

과정이라는 생각이 들어서, 앞으로 이 방향으로 더 깊이 탐구해보고 싶다.

이*운 : '오늘부터 개발자'를 읽으며 개발자라는 직업에 대해 구체적으로 들여다보았다. 프론트엔드, 백엔드, 데이터, 데브옵스 등 직군의 스펙트럼을 확인하면서 내가 어느 방향으로 나아가고 싶은지 생각해볼 수 있었다. 또한 토론을 통해 개발자로서 어떤 태도를 가져야 하는지를 고민하게 되었다.

김*비 : 『오늘부터 개발자』를 읽으면서, 그동안 내가 개발자라는 직업에 대해 생각보다 많은 것을 모르고 있었다는 점을 깨달았다. 프론트엔드와 백엔드와 같은 기본적인 개념조차 명확히 이해하지 못하고 있었는데, 이 책을 통해 개발자의 역할과 분야를 보다 체계적으로 이해할 수 있게 되었다. 특히 '데브옵스(DevOps)'라는 직군을 처음 알게 되었는데, 개발자라는 직업 안에도 생각보다 다양한 분야가 존재한다는 점이 인상 깊었다. 이를 계기로 나는 앞으로 어떤 방향의 개발자가 되고 싶은지 고민해보는 시간을 가질 수 있었다. 또한 저자가 항상 노트북을 들고 다니며 문제가 발생했을 때 즉시 해결하려는 태도를 보며, 이것이 개발자의 기본적인 자세일 수도 있겠다는 생각이 들었다. 개발자는 끊임없이 배우고 성장해야 하는 직업이라는 점에도 깊이 공감하게 되었고, 나 역시 더 꾸준히 공부해야겠다는 다짐을 하게 되었다. 한편, 책에서는 개발자 수요가 꾸준히 증가하고 있다고 언급하지만, 최근 AI 기술의 발전으로 상황이 변화하고 있다는 점도 느낄 수 있었다. 실제로 AI가 단순한 코드 작성 업무를 상당 부분 대체하고 있으며, 관련 일자리가 감소했다는 뉴스를 접하면서 개발자의 미래에 대해 다시 생각해보게 되었다. 이러한 변화 속에서 뒤처지지 않기 위해서는 단순한 코딩 능력을 넘어서, 더 높은 수준의 문제 해결 능력을 갖춘 개발자가 되어야 한다고 느꼈다.

황*민 : 이번 독서를 통하여 개발자의 여러가지 직군들에 대해 알게되며 막연하기만 했던 진로가 명확해지게 되었다 또한 그 전까지는 개발자에게 가장 중요한 것은 개발하는 것인줄 알았지만 이 책을 통해 유지보수의 중요성에 대해 다시 한번 알게되는 계기가 되었으며 개발자에게 필요한 역량에 대해 토론해 볼 수 있는 계기가 될 수 있었다. 이번 독서를 통해 앞으로는 수업으로 배우는 내용 이외에도 더 새로운 내용들을 공부해 봐야겠다는 생각을 가지게 되었다.

- 핵심 주제 1. 개발자에게 중요한 역량은 무엇인가?

박*진 : 꾸준히 새로운 것을 받아들이고 공부하는 자세가 가장 중요하다고 생각한다. 특히 개발 직군은 새로운 기술이 빠르게 등장하기 때문에 변화에 유연하게 적응하는 능력이 핵심이라고 느꼈다. 꾸준함이 중요한 이유는 개발은 단기간에 익힐 수 있는 분야가 아니기 때문이다. 하루하루 조금씩 쌓아온 지식과 경험이 결국 큰 차이를 만들고, 그 축적이 실력이 된다고 생각한다.

또한 요즘의 개발자에게는 자신이 작성한 코드를 스스로 이해하고, 다른 사람도 이해할 수 있도록 주석화, 문서화하는 능력이 중요하다고 생각한다. AI가 코드를 생성해주는 시대에는 오히려 그 코드를 읽고 설명할 수 있는 능력이 차별점이 되기 때문이다.

이*운 : 코티원들과 이야기를 나누면서 '문제를 정의하는 능력이 중요한가, 코드를 깔끔하게 짜는 능력이 더 중요한가'라는 질문이 머릿속에 남았다. 개인적으로는 두 가지가 다른 층위에 있다고 생각한다. 코드를 잘 짜는 것은 기본기이기도 하며 AI의 도움을 가장 많이 받을 수 있는 분야이다. 문제를 정의하는 시각은 그 위에서 발휘되는 능력이다. 이러한 역량을 개발하기 위해서는 여러 사회적 문제에 관심을 가지고 문제를 정의하고 해결 방법을 고민해보는 훈련도 도움이 될 것이다. 또한 AI 저작권이나 윤리 문제가 대두되고 있지만 명확한 해결책은 없는 이 시점에서 우리 개개인은 각자만의 확고한 윤리관을 가져야 할 것이다.

김*비 : 개발자에게 가장 중요한 역량은 디버깅 능력과 문제 해결 능력이라고 생각한다. 코딩을 하다 보면 다양한 오류가 발생하게 되는데, 일부는 AI의 도움으로 해결할 수 있지만, 여전히 사람이 직접 원인을 분석하고 해결해야 하는 문제들도 존재한다. 이러한 상황에서 오류의 원인을 정확히 파악하고 해결하는 능력은 개발자에게 필수적인 역량이라고 할 수 있다. 따라서 단순히 코드를 작성하는 능력보다, 문제를 분석하고 해결하는 능력이 더 중요하다고 생각한다.

황*민 : 개발자는 문제를 정의 할 수 있는 능력이 중요하다 생각한다. 현재 AI의 도움을 통해 코드를 짤수있는 시대가 찾아왔지만 AI에게 코드를 짜게 맡기려면 자신이 해결해야하는 문제를 정의하고 그걸 프롬프트를 통해 자신의 의도를 전달해야 한다. 그렇기 때문에 시게 코드를 맡기더라도 결국 문제를 정확하게 인식하고 정의하여 전달할 수 있어야 한다는 것이다. 그렇지만 자신(또는 AI)가 짠 코드를 이해하고 다른이들과 작업을 공유하기위해 주석화 하는것도 중요한 부분이다. 또한 개발자는 이제 윤리적 문제에서 멀어질 수 없다 생각한다. 지금까지는 기술 개발이라는 이유로 윤리적 문제에서 벗어나 학습에 사용된 데이터의 저작권 침해와 AI가 생성한 자료의 저작권등 여러가지 윤리적 문제가 대두되고 있기에 기술개발보다 이러한 윤리적 문제 해결이 우선되어야 하지않을까 생각한다.

- 핵심 주제 2. 개발 진입 장벽 하락, 위기인가 기회인가?

박*진 : 위기라고 생각한다. 개발의 진입장벽은 낮아졌지만, 신입 개발자로서의 취업 장벽은 오히려 높아졌다고 느낀다. AI의 도움으로 누구나 코드를 작성할 수 있게 된 지금, 단순히 "코딩을 할 줄 안다"는 것만으로는 더 이상 경쟁력이 되지 않기 때문이다. 오히려 그만큼 개발자에게 요구되는 것들이 더 다양해지고 높아졌다. 실제로 최근 채용공고들을 보면 단순 구현 능력에 더하여 문제를 스스로 정의하고 해결하는 능력, 팀원과 원활하게 소통하고 협업하는 능력까지 갖춘 사람을 원하는 경우가 눈에 띄게 많아졌다. 결국 개발이라는 분야에 발을 들이는 것은 쉬워졌지만, 그 안에서 살아남는 것은 오히려 더 어려워진 시대가 된 것 같다.

이*운 : 올해 1분기 미국 IT 일자리가 5만 개 이상 감소했다는 뉴스를 보면 우리 직업은 위기에 놓인 것 같다. 기업의 입장에서는 소수의 인원으로 AI를 활용하여 발전할 수 있지만 남은 사람들은 어떠한 방식으로 자신의 경쟁력을 높일 수 있을 것인지 고민해볼 필요가 있다. AI가 만든 결과물을 검증하고 수정하고 책임지는 사람은 개발자여야 하지만 그러한 일은 이미 근무중인 시니어 개발자만 있어도 충분하지 않은가? 단순히 개발만 잘하는 사람보다는 복합적인 인재가 되어야 할 것 같다.

김*비 : 나에게서는 위기와 기회가 동시에 존재한다고 생각한다. AI의 발전으로 인해 개발 진입 장벽이 낮아지고, 더 많은 사람들이 개발에 접근할 수 있게 되었다. 이는 긍정적인 변화이지만, 동시에 경쟁이 심화된다는 점에서는 위기로 작용할 수 있다. 특히 단순한 업무는 AI로 대체되는 상황에서, 차별화된 역량을 갖추지 못한다면 경쟁에서 뒤처질 가능성이 높다. 하지만 이러한 상황을 부정적으로만 볼 수는 없다. 오히려 꾸준한 학습과 노력을 통해 자신의 역량을 강화한다면, 더 높은 수준의 개발자로 성장할 수 있는 기회가 될 수도 있다고 생각한다. 결국 중요한 것은 환경이 아니라, 그 속에서 얼마나 성장하려는 자세를 가지느냐라고 느꼈다.

황*민 : 개발자의 진입장벽이 낮아지며 더 많은 인력들이 개발 업계로 뛰어들게 되었다. 하지만 현재 기업들은 AI를 통해 개발 시간을 단축하고 인력을 감축할 수 있기에 책임 및 AI의 코드를 검증하고 책임질 시니어 개발자 위주로 채용하고 신입 개발자는 인력축소를 하는 중이다. 이러한 상황으로 취업장벽은 높고 진입장벽이 낮음으로 신입개발자로 취업시장에 뛰어들 우리들은 경쟁력을 높이기 위해 소통능력과 협업능력등을 모두 지닌 인재가 되어야 한다고 생각한다.

- 핵심 주제 3. 전공자만의 강점은 무엇인가?

박*진 : 전공자에게는 프로젝트 기회, 수업의 기회 등 비전공자에 비해 기회의 문이 더 넓게 열려있다고 생각한다. 비전공자는 외부 프로그램을 통해서만 경력을 쌓고 공부를 할 수 있지만 전공자는 전공 수업 및 전공과 연결되는 다양한 프로그램을 통해 비교적 체계적으로 프로젝트 경험과 지식을 쌓을 수 있다. 또한 전공 수업에서 배우는 컴퓨터 구조, 알고리즘, 자료구조 같은 CS 기초 지식은 단순히 코딩 툴을 배우는 것과는 깊이가 다르다.

이*운 : 공학적인 원리를 이해하고 있는 사람은 새로운 언어나 프레임워크를 배울 때 속도가 다르다고 한다. 또한 4년제 전공자로서 부트캠프나 비전공자에 비해 차별점은 분명히 존재한다. 또한 학교를 다니면서 오는 기회의 폭이 넓다. 동아리, 팀 프로젝트, 인턴, 학부 연구생, 박람회, 해커톤 등의 기회에 적극적으로 참여해야겠다고 생각했다.

김*비 : 전공자의 가장 큰 강점은 자신의 아이디어를 실제로 구현할 수 있는 능력이라고 생각한다. 책에서도 언급되었듯이, 단순히 아이디어를 떠올리는 것에 그치지 않고 이를 직접 앱이나 게임으로 만들어낼 수 있다는 점은 매우 큰 장점이다. 이러한 과정은 단순한 기술 습득을 넘어 큰 즐거움과 성취감을 제공한다. 또한 자신이 만든 결과물이 실제로 사람들에게 사용되고 긍정적인 반응을 얻는다면, 그 경험은 개발자로서의 동기 부여에도 큰 영향을 미칠 것이라고 생각한다.

황*민 : 전공자는 비전공자보다 기본기를 더 쌓을 수 있고 더 많은 기회를 얻을 수 있다고 생각한다. 비전공자가 코딩을 배울때는 단순히 프로그래밍 언어를 사용하는 것에 대해서만 배우는 경우가 많다 하지만 전공자는 프로그래밍 언어를 배우며 자료구조나 알고리즘, 컴퓨터 시스템 등 기본 CS지식을 배우기 때문에 이러한 CS지식을 바탕으로 개발 역량을 늘릴 수 있게 된다 또한 전공자는 대학을 다니며 동아리나 프로젝트 등의 여러가지 기회들이 주어지기 때문에 이러한 전공자의 강점들을 살리기 위해 열심히 대학생활에 임해야 한다는 생각이 들었다.

- 핵심 주제 4. 개발자의 핵심 능력 - 유지보수

박*진 : 에러를 해석하고 해결하는 능력이 핵심이 될 것 같다. 초기에 코드를 작성하는 것은 AI의 도움을 받아 누구나

쉽게 시작할 수 있는 일이 되었다. 하지만 그 코드를 실제 상황에 적용했을 때 발생하는 예상치 못한 에러를 이해하고 해결하는 것은 전혀 다른 이야기이다. 이러한 경험은 직접 부딪혀보지 않으면 쌓을 수 없는 것이다. 앞으로는 단순히 코드가 돌아가는 것에 만족하지 않고 왜 에러가 났는지, 어떻게 해결할 수 있는지를 끝까지 파고드는 습관을 들여야겠다

이*운 : 짜여진 코드를 이해할 수 있도록 주석과 문서화 작업을 잘 해야할 것 같다. 앞으로 프로젝트를 할 때 다른 사람이 봐도 이해할 수 있는, 6개월 뒤의 내가 이해할 수 있는 코드를 작성할 수 있도록 주의를 기울여야겠다.

김*비 : 개발자의 중요한 역량 중 하나는 유지보수 능력이라고 생각한다. 이는 앞서 언급한 디버깅 능력과도 밀접하게 연결된다. 코드에서 오류가 발생했을 때, 문제의 원인을 빠르게 파악하고 수정할 수 있는 능력은 매우 중요하다. 이를 위해서는 평소에 코드를 체계적으로 작성하고, 주석을 통해 코드의 의도와 목적을 명확히 기록하는 습관이 필요하다. 주석은 단순한 설명을 넘어서, 나중에 코드를 다시 이해하고 수정하는 데 큰 도움이 된다. 또한 저자가 자신의 프로젝트를 블로그나 깃허브에 꾸준히 정리하는 습관을 강조한 점도 인상 깊었다. 이러한 기록은 자신의 성장을 확인할 수 있을 뿐만 아니라, 다른 사람과의 협업에서도 중요한 역할을 한다고 생각한다. 나 역시 이러한 습관을 본받아야겠다고 느꼈다.

황*민 : 코드를 이해할 수 있는 능력 그리고 코드를 이해할 수 있도록 주석을 쓰고 문서화 하는 능력이 중요하다 생각한다. 개발자로 일하면 여러 사람과 협업하기도 하고 자신이 짜놓은 코드를 타인이 수정할 일도 생긴다. 그런 상황에서 주석이 존재하지 않고 관련 문서도 없다면 다른사람들 그리고 몇개월 후의 자신조차 해석할 수 없게 된다. 주석을 쓰는 것은 개발자가 자신이 짠 코드의 책임을 지는것이라 생각한다.

- 핵심 주제 5. 평생 공부해야 하는 직업, 장점인가 단점인가?

박*진 : 장점이라고 생각한다. 개인적으로 가장 중요하게 여기는 가치는 '성장'이다. 성장하지 못하고 제자리에 머무르는 느낌이 들 때 무기력함을 느끼곤 한다. 개발자라는 직업은 평생 새로운 것을 배우고 성장해야 하는 직업이기 때문에 오히려 그 과정 자체가 무기력함을 이겨내고 삶의 원동력이 될 수 있다고 생각한다. 물론 끊임없이 공부해야 한다는 부담감이 없는 건 아니다. 하지만 그것이 삶의 원동력이 된다면 충분히 감수할 수 있는 부분이라고 느꼈다.

이*운 : 내가 삶에서 가장 중요하다고 생각하는 것은 경험, 성장, 성취다. 이러한 관점에서 개발자는 삶의 가치를 끊임없이 검증해내는 직업이다. 기술 변화 속도가 너무 빠르다는 부담감도 있지만, 배움이 멈추지 않는다는 것 자체는 큰 동력이 된다. 배움의 부재가 인지 기능 저하로 이어진다는 이야기처럼, 이 직업의 특성이 장기적으로는 삶의 질을 높여 줄 수도 있을 것이라고 생각한다.

김*비 : 나는 솔직히 공부를 좋아하는 편은 아니다. 그래서 개발자라는 직업이 나에게 잘 맞는지에 대해 고민해본 적도 있다. 하지만 개발자는 끊임없이 배우고 성장해야 하는 직업이며, 이러한 과정 자체는 분명 긍정적인 의미를 가진다고 생각한다. 사람은 성장하기 위해 지속적으로 배우고 경험해야 하며, 그런 점에서 평생 공부하는 것은 하나의 장점이라고 볼 수 있다. 물론 공부를 좋아하지 않는 나에게는 부담으로 느껴질 수도 있지만, 만약 이 분야에 대한 흥미를 더 키운다면 자연스럽게 즐겁게 공부할 수 있을 것이라고 생각한다.

황*민 : 흔히들 개발자는 평생 공부하는 직업이라 말한다. 물론 어떤 직업이든 끊임없이 공부해야 하지만 개발자는 특히 새로운 기술들을 끊임없이 배우고 학습해야만 타인에게 대체당하지 않을 수 있다. 하지만 이때문에 개발자는 자신이 노력이 더 드러나는 직업이라 생각한다. 내가 한 노력만큼 내가 성장하는게 눈에 보인다는것은 삶을 살아가는데 큰 원동력이 될것이라 생각한다.

활동 자료

북클럽 팀명	NextDev					
일시/장소	일시	2026.04.14			장소	도서관 3층
참여 인원	학과	빅데이터	학번	2024****	이름	박*진
	학과	컨텐츠IT	학번	2024****	이름	이*운
	학과	스마트IoT	학번	2024****	이름	김*비
	학과	소프트웨어 AI 계열	학번	2026****	이름	황*민
	참석인원(명)	4 / 4		참가자 대표	박*진 (서명)	
도서	도서명	개발자 원칙(확장판)			저자	박성철 외 9
	출판사	골든래빗	출판연도	2022	번역자	
오늘의 북클럽 주제	AI 시대 개발자					
활동 내용 & 방법						
<p>책을 읽은 후, 인상 깊었던 내용을 나누기 전에 먼저 각자 책 소감을 자유롭게 이야기하는 시간을 가졌다. 9명의 현직 개발자가 정의한 '좋은 개발자'의 기준, 내재 동기과 외부 보상에 관한 구절, 그리고 소통 능력의 중요성 등 각자가 인상 깊게 읽은 부분을 공유하며, 앞으로 어떤 방향의 개발자가 되고 싶은지에 대한 생각을 나눴다. 이후 사전에 준비한 핵심 주제를 바탕으로 본격적인 토론을 진행했다. 주요 토론 주제는 다음과 같았다.</p> <p>첫 번째로 'AI 시대에 더욱 중요해지는 설계 능력, 어떻게 키울 수 있을까?'를 논의하며, AI가 코드를 자동 생성해주는 시대일수록 설계 능력이 핵심 역량이 된다는 점에 공감했다. 기록과 수정의 반복, 다이어그램을 활용한 시각적 설계, 직접 프로젝트를 해보는 경험, 그리고 타인에게 설명할 수 있을 정도로 설계를 이해하는 것이 중요하다는 이야기를 나눴다.</p> <p>두 번째로 'AI 시대에도 좋은 개발자의 기준은 달라졌는가, 같은가?'를 주제로 토론했다. 설계력, 오류 해결 능력, 커뮤니케이션 능력 등 기준 자체는 크게 변하지 않았으나, AI로 인해 단순 구현에 드는 시간이 줄어든 만큼 문제를 정의하는 능력과 협업 능력이 더욱 부각되고 있다는 점, 그리고 다양한 기술 분야를 아우를 수 있는 유연성이 중요해졌다는 의견을 나눴다.</p> <p>세 번째로 '배운 것을 기록하고 공개하는 습관, 얼마나 중요한가?'를 논의하며, 오류를 만났을 때 단순히 해결하고 넘어가는 것이 아니라 그 과정을 기록함으로써 사고를 객관화하고 포트폴리오로도 활용할 수 있다는 점을 공유했다. 기록을 공개하면 타인의 피드백을 통해 스스로의 논리를 검증할 수 있고, 다른 개발자와 학습자에게도 도움이 될 수 있다는 점에서 의미 있는 습관임을 확인했다.</p> <p>네 번째로 '개발자로서 자존감을 키우는 방법은 무엇인가?'를 주제로, 타인과의 과도한 비교보다는 어제의 나와 비교하며 성장하는 것이 중요하다는 데 의견을 모았다. 직접 프로젝트를 완성하면서 느끼는 성취감, 외부 피드백을 공격이 아닌 정보로 객관적으로 받아들이는 태도, 그리고 결과물을 외부에 공개하고 기록으로 남기는 과정이 자존감의 토대가 된다는 이야기를 나눴다.</p>						
활동 결과						
<p>- 책소감</p> <p>박*진: 이 책에서 가장 인상 깊었던 부분은 서두에 등장하는 9명의 개발자가 정의한 '좋은 개발자'였다. 각자 정의는 달랐지만 결국 그 정의들은 일 잘하는 좋은 사람이라고 생각한다. 소통하고, 문제를 해결하고, 함께하는 사람을 배려하는 것 등 이는 개발자만의 덕목이 아니라 어느 분야에서든 요구되는 자질이다. 나 역시 좋은 개발자이기 이전에 함께 일하고</p>						

싶은 사람이 되는 것을 먼저 목표로 삼고자 한다.

이*운: 개발자 원칙은 9명의 현직자들이 각자의 개발 원칙을 하나씩 풀어내는 방식이라 읽는 내내 선배들의 조언을 듣는 것처럼 느껴졌다. 가장 인상 깊었던 것은 개발자의 정의였다. 주어진 문제를 잘 푸는 것을 넘어 이 문제가 정말 풀어야 할 문제인지 의심할 줄 아는 사람, 가치를 더하는 사람 등 내가 앞으로 어떤 사람이 되어야 할 것인지 방향을 잡는데 도움이 되었다.

김*비: 개발자 원칙 내용 속 "내재 동기를 개발하고 고양하는 데 관심이 있는 사람이라면 금전적인 보상처럼 외부에서 통제되는 체제에 집중해서는 안 된다- 에드워드 디씨"라는 말이 깊이 와닿았다. 나는 어렸을 때부터 코딩을 좋아해서 개발자를 목표로 달려온 사람이다. 그러나 요즘 슬럼프가 온 것인지 코딩이 재미없게 느껴지면서, 내가 이 길로 가는 것이 맞는지 고민하게 되었다. 좋은 개발자가 된다면 분명 돈을 잘 벌 수 있을 것이다. 하지만 내가 즐기지 않는다면 오래 버티지 못할 것 같다는 생각이 들었다. 그래서 다시 코딩을 즐겁게 하던 때로 돌아가고 싶다는 마음이 들었고, 이 책을 읽으며 많은 생각을 하게 되었다.

황*민: 개발자 원칙을 읽으며 가장 인상 깊었던 부분은 앞부분의 현직자들의 좋은 개발자의 정의였다. 각자가 정의한 원칙들은 서로 달랐지만 가장 강조된 부분은 '소통능력' 이라고 생각한다. 개발을 잘 하는 것도 물론 중요한 능력이지만 결국 일 이기에 타인과의 의사소통 능력이 필수적이라는 것이다. 또한 이제 점점 개발자들이 개발의 전 분야에 인공지능의 활용되면서 인공지능과의 협업을 위한 언어적 능력 또한 길러야 한다는 생각이 들게 되었다.

- 핵심 주제1. AI 시대에 더욱 중요해지는 설계 능력, 어떻게 키울 수 있을까?

박*진: AI가 코드를 자동으로 생성해주는 시대일수록, 설계 능력의 중요성은 커진다. 설계는 단순히 코드를 짜기 전의 준비 단계가 아니라 문제를 구조화하는 사고라고 생각한다. 기록하고 수정하는 반복, 그림으로 먼저 이해하는 습관 등 이러한 방식들은 사고를 외부화하여 논리적 허점을 스스로 발견하게 만드는 과정이다. AI에게 좋은 요청을 하기 위해서는 먼저 스스로 문제를 설계할 수 있어야 한다고 생각한다.

이*운: 개발 능력이 좋아지면 설계 능력은 저절로 따라올 것이라고 생각하긴 한다. 코드를 많이 짜본 사람일수록 이렇게 짜면 나중에 문제가 생기겠다는 감각이 생기기 때문이다. 하지만 책을 읽으면서 그것만으로는 부족할 수 있겠다는 생각도 들었다. 경험이 쌓이더라도 의식적으로 설계를 고민하지 않으면 그냥 익숙한 방식을 반복하는 것에 그칠 수 있기 때문이다.

책에서는 AI를 쓰면 쓸수록 오히려 더 깊고 제대로 된 학습이 필요해진다는 말이 나온다. AI가 코드를 생성해줄 수는 있지만, 그 코드가 좋은 설계인지 나쁜 설계인지 판단하는 건 결국 나의 몫이다. 그래서 나는 코드를 짰 후에 이걸 처음 보는 사람에게 설명할 수 있는가를 스스로 물어보는 습관을 들여보려 한다.

김*비: 우선 프로젝트나 개발을 하면서 기록을 하는 것이 중요하다고 생각한다. 기록을 통해 과거에 내가 어떤 방식으로 개발하고 설계를 했는지 돌아볼 수 있고, 부족한 점을 보완하며 설계 능력을 키울 수 있다. 또한 설계를 잘하는 다른 사람이 작성한 코드나 기록을 보며 배우는 것도 좋은 방법이다. 무엇보다 중요한 것은 경험이라고 생각한다. 많이 설계해보고 직접 개발해보는 과정에서 실력이 쌓이기 때문이다.

또한 토론에서 나온 것처럼 다이어그램이나 그림을 활용해 설계하는 방법도 효과적이라고 생각한다. 기록은 글로만 해야 한다는 고정관념에서 벗어나, 그림으로 표현하고 구조를 시각적으로 이해하는 것이 도움이 된다. 그리고 내가 다른 사람에게 설명할 수 있을 정도로 설계를 이해한다면 더 좋은 설계를 할 수 있을 것이라고 생각한다.

황*민: AI시대가 찾아오며 중요해지는 설계 능력을 키우기 위해 직접 프로젝트를 설계하고 짜보고 이를 공유하며 평가 받는 과정들을 통해서 설계능력을 키울 수 있을 것 이라 생각한다. 또한 AI가 코드를 짤 수 있는 시대이기에 코드를 이해하고 설계하여 자신의 목적을 AI에게 설명하고 AI가 짠 코드를 자신의 의도에 맞게 변형해야 하기 때문이다

- 핵심 주제2. AI 시대에도 '좋은 개발자'의 기준은 달라졌는가, 같은가?

박*진: 기준 자체는 달라지지 않았다. 다만 그 기준이 더욱 선명해졌다.

설계력, 오류 해결 능력, 다양한 분야를 소화하는 유연성 등 이것들은 AI 이전에도 중요했지만, AI가 단순 구현을 담당하게 되면서 이제는 이 능력들이 개발자의 핵심 역량으로 부각된다. 특히 커뮤니케이션 능력은 개발자 간의 소통을 넘어, 비개발자와의 협업에서 더 크게 요구된다는 의견이 인상적이었다.

이*운: AI 시대에도 책의 저자들이 이야기한 좋은 개발자의 기준은 거의 같다고 생각한다. AI를 활용하여 소프트웨어에

가치를 더할 수 있으며, 개발 실력 뿐 아니라 커뮤니케이션 능력 또한 중요하다. 오히려 AI 덕분에 단순 구현에 드는 시간이 줄어든만큼 사람과 사람 사이의 협업 능력과 문제를 정의하는 능력이 더 부각되는 시대가 됐다고 느낀다.

취업 시장도 달라질 것이다. 기업에서는 지원자의 코딩 능력만 보는 것이 아니라 AI 활용 능력이나 기타 커뮤니케이션 능력 등을 볼 수 있는 면접을 준비해야 할 것이다. 나아가 지원자 스스로도 개발 능력 외에 디자인이나 기획 등의 분야까지 능력을 확장할 수 있어야 한다. AI 덕분에 개발에 드는 시간이 단축된 만큼 그 여백을 다른 역량을 키우는 데 쓸 수 있기 때문이다.

김*비: AI가 점점 발전하면서 기존에 사람이 하던 일을 대신하게 되고, 그로 인해 인력이 줄어드는 상황이 나타나고 있다. 예전에는 개발자에게 커뮤니케이션 능력이 매우 중요한 역량으로 강조되었지만, 현재는 소규모 팀이 많아지면서 예전만큼 중요한지에 대해서는 의문이 들기도 한다. 물론 어떤 팀이든 혼자가 아니라면 커뮤니케이션 능력은 여전히 중요하다.

하지만 현재의 AI 시대에서는 커뮤니케이션 능력뿐만 아니라, 실질적인 능력이 뛰어난 사람이 더 중요한 개발자라고 생각한다. 특히 다양한 기술을 다룰 수 있는 사람이 더 유리하다고 느낀다. AI로 인해 인력은 줄어들고, 적은 인원이 여러 역할을 맡는 경우가 많아지고 있기 때문이다. 이러한 환경에서는 코딩뿐만 아니라 디자인, 설계 등 다양한 역량을 갖춘 사람이 좋은 개발자가 될 것이라고 생각한다.

황*민: 달라지지 않았다 라고 생각한다. AI시대가 찾아오면서 한 개발자가 여러 개발분야를 혼자 책임지는 경우가 많아 졌지만 그 와중에서도 변하지 않는 가치는 의사소통 능력이다. 자신이 개발분야를 모두 책임지게 되더라도 결국 다른 팀과의 소통 다른 분야와의 소통 또한 AI와의 소통을 위해서도 의사소통능력은 언제나 좋은 개발자의 기준이 될 것이라 생각한다.

- 핵심 주제3. 배운 것을 기록하고 공개하는 습관, 얼마나 중요한가?

박*진: 기록은 단순한 메모가 아니다.

틀렸던 것, 오래 고민한 것, 해결에 이르는 과정을 남기는 행위는 자신의 사고를 객관화 하는 훈련이라고 생각한다. 이를 공개하는 것은 타인의 시선을 통해 스스로의 논리를 검증하는 기회가 된다. 앞으로는 틀린 것, 막혔던 것 등을 기록하고 공개하는 습관을 들여보고자 한다. 틀린 것을 공개한다는 것이 다소 부끄럽기도 하지만 부끄러움만 극복한다면 실력을 향상시킬 수 있을 것이라고 확신한다.

이*운: 오류를 만났을 때 단순히 해결하고 넘어가는 것이 아니라, 왜 그 오류가 생겼는지 확인하고 내 것으로 만드는 과정이 진짜 성장이다. 나는 어려운 문제가 있거나 기억하고 싶은 오류를 만나면 문제를 해결한 뒤 깃허브에 기록하는 습관을 들이려 하고 있다. 기록은 기억을 더 오래 지속하게 만들어주고 타인에게 보여질 수도 있다는 생각 때문에 더 꼼꼼하게 임하게 되는 장점도 있다.

김*비: 심비우스 활동을 하면서 개발자 관련 책을 여러 권 읽었는데, 공통적으로 강조하는 내용이 바로 기록이었다. 이 책에서도 기록의 중요성을 강조하고 있으며, 그만큼 개발 과정에서 기록은 빠질 수 없는 요소라고 생각한다. 기록을 하면서 배우고 성장할 수 있을 뿐만 아니라, 내가 만든 결과물을 포트폴리오로 활용할 수도 있다.

또한 내가 작성한 기록을 공개하면 다른 개발자나 학습자들에게도 도움이 될 수 있다. 나 역시 공부하면서 많은 도움을 받았기 때문에, 앞으로는 나도 다른 사람에게 도움을 줄 수 있는 사람이 되고 싶다. 기록하고 공개하는 습관은 나를 더 나은 개발자로 성장시키는 데 큰 역할을 할 것이라고 생각한다.

황*민: 내 능력을 기를 수 있는 가장 간단한 방법이라 생각한다.

내가 배운것을 기록하고 이를 공개하는 습관을 가진다면 자신이 한 작업물과 자신이 배운 내용을 타인과 공유할 수 있게 되며 그 공유를 통해 자신의 학습 내용을 타인에게 평가 받을 수 있고 여러 사람의 의견을 얻을 수 있게 되며 이를 통해 자신의 제작물을 더욱 발전시키며 앞으로의 개발에서의 자신만의 기준을 세울 수 있게 된다 생각한다.

-핵심 주제4. 개발자로서 자존감을 키우는 방법은 무엇인가?

박*진: 어느 분야든 타인의 성과는 눈에 띄고 나의 부족함은 크게 느껴진다. 때문에 비교에 의해 자존감이 많이 떨어지고 일에 대한 의욕을 잃을 수 있다.

자존감을 키우기 위해서는 무엇보다 외부 피드백을 객관적으로 받아들이는 것이 중요하다. 피드백을 공격이 아닌 정보로 객관적으로 처리할 수 있다면, 내가 실제로 무엇을 개선해야 하는지 명확히 파악하고 성장 방향을 바르게 잡을 수

있다. 또한 타인과 비교하는 것이 아닌 어제의 나와 비교하며 성장하고, 스스로 확인할 수 있는 결과물을 만들어 기록하면서 성취 경험을 쌓는 것이 중요하다. 객관적으로 받아들인 피드백은 어제의 나와 비교하는 기준으로 활용할 수 있어, 자존감을 지키면서도 꾸준히 성장할 수 있는 원동력이 된다.

이를 위해 앞서 토론한 것처럼 자신의 결과물을 외부에 공유하고 정리하는 과정이 중요하다고 생각한다. 결과물을 외부에 공개하는 것은 객관적인 피드백을 얻을 수 있는 기회가 됨과 동시에, 내가 쌓아온 성장을 눈에 보이는 형태로 남기는 일이기도 하다. 이러한 기록이 쌓일수록 스스로의 성장을 실감할 수 있고, 그것이 곧 자존감의 토대가 된다.

이*운: 챗터2에서 인용된 구절인"배우는 일, 그것은 즐겁다. 생각하는 일은 더 즐겁다. 창조하는 인생이야말로 최고의 인생이다." 이 문장이 개발자 자존감의 본질을 담고 있다고 생각한다. 자존감은 남과의 비교에서 오는 것이 아니다. 어제의 나보다 나아졌는가, 오늘 또 하나 작은 일을 성취해냈는가. 점점 성장해나가는 나를 보며 채워지는 것이다. 개발이든 취미든 여러가지 분야에서 나보다 잘하는 사람들은 항상 존재한다. 그들을 불안의 원인으로 볼 것이 아니라 성장의 원동력으로 보는 시각의 전환이 필요하다. 잘하는 사람들 사이에 있다는 것은 불안의 신호가 아니라 성장의 조건이다.

김*비: 자존감이 떨어질 때는 보통 다른 사람과 나를 비교할 때가 많다. 내가 다른 사람보다 부족하다고 느끼면 자연스럽게 자신감도 낮아진다. 그래서 가장 먼저 해야 할 것은 과도한 비교를 멈추는 것이라고 생각한다. 적당한 비교는 동기부여가 될 수 있지만, 지나친 비교는 오히려 자신을 힘들게 만든다.

나는 실제로 프로젝트를 진행하면서 자존감이 높아진 경험이 있다. 평소에는 내가 할 줄 아는 것이 별로 없다고 생각했고, 주변 동기들보다 부족하다고 느꼈다. 하지만 직접 프로젝트를 해보면서 나도 충분히 개발을 할 수 있는 사람이라는 것을 깨닫게 되었고, 내 수준을 다시 바라보게 되었다.

따라서 자신이 할 수 있는 개발을 직접 해보고, 결과물을 완성하면서 느끼는 성취감이 자존감을 높이는 데 큰 도움이 된다고 생각한다.

황*민: 개발자로서의 자존감을 키우려면 일단 자기자신을 너무 자책하는 것을 멈추어야 한다 생각한다. 과한 자신감도 독이 되지만 과한 자책은 자기자신을 발전시키지 못한다. 이를 해결하기 위해 자신의 결과물과 학습내용을 공개함으로써 자존감을 키울 수 있을 것이라 생각하였다.

활동 자료

--

북클럽 팀명	NextDev					
일시/장소	일시	2026.04.28			장소	도서관 3층
참여 인원	학과	빅데이터	학번	2024*****	이름	박*진
	학과	컨텐츠IT	학번	2024*****	이름	이*운
	학과	스마트IoT	학번	2024*****	이름	김*비
	학과	소프트웨어 AI 계열	학번	2026*****	이름	황*민
	참석인원(명)	4 / 4			참가자 대표	박*진 (서명)
도서	도서명	바이브 코딩 너머 개발자 생존법			저자	애디 오스마니
	출판사	한빛미디어	출판연도	2025	번역자	강민혁
오늘의 북클럽 주제		SI가 코드를 짜주는 시대				
활동 내용 & 방법						
<p>책을 읽은 후, 인상 깊었던 내용을 나누기 전에 먼저 각자 책 소감을 자유롭게 이야기하는 시간을 가졌다. SI가 코드를 대신 짜주는 시대에 개발자란 무엇인가에 대한 생각, 기초 없이 SI를 사용하면 결과를 검증할 기준 자체가 없어진다는 점, 그리고 개발자의 역할이 사라지는 것이 아니라 더 높은 수준의 판단과 이해를 요구하는 방향으로 변화하고 있다는 점 등 각자가 인상 깊게 읽은 부분을 공유하며, 앞으로 어떤 방향의 개발자가 되고 싶은지에 대한 생각을 나눴다. 이후 사전에 준비한 핵심 주제를 바탕으로 본격적인 토론을 진행했다. 주요 토론 주제는 다음과 같았다.</p> <p>첫 번째로 'SI 사용 사실, 어디까지 공개해야 할까?'를 논의하며, AI 사용 여부 자체는 반드시 밝혀야 한다는 데 팀원 모두 동의했다. 다만 프롬프트 공개에 대해서는 의견이 갈렸다. 프롬프트는 단순한 입력값이 아니라 문제를 어떻게 정의하고 접근하는지를 담은 개인의 사고 과정이자 노하우이며, 이를 구성하는 능력 자체가 역량의 일부라는 시각에서 전면 공개에는 반대 의견이 많았다. 공개의 목적은 투명성 확보이고, 사용 여부 표기만으로도 그 목적은 충분히 달성된다는 결론에 이르렀다.</p> <p>두 번째로 '코드를 승인만 하는 개발자도 개발자인가?'를 주제로 토론했다. 팀원 전원이 '그렇다'는 입장을 취하되, 조건을 달았다. 단순히 승인 버튼을 누르는 행위가 아닌, 해당 코드를 이해하고 적합성과 문제점을 판단하여 책임지는 과정이 수반되어야 한다는 데 의견이 모였다. 코드를 직접 작성하는 것보다 결과물을 온전히 이해하고 책임지는 능력이 개발자의 본질이라는 점, 그리고 AI 시대에 프로그래머의 역할은 축소될 수 있어도 개발자는 오히려 더 중요해진다는 이야기를 나눴다.</p> <p>세 번째로 '기초 없이 SI를 쓰면 독이 된다 - 그렇다면 기초란 무엇인가?'를 논의하며, 기초의 정의에 대해 팀원별로 다양한 견해가 나왔다. 자료구조, 알고리즘 등 CS 지식도 중요하지만, 특정 언어나 프레임워크를 넘어 코드 앞에서 '왜?'라는 질문을 멈추지 않는 태도와 사고방식 자체가 진정한 기초라는 의견이 중심을 이루었다. SI가 생성한 코드를 그대로 수용하지 않고 스스로 이해하고 검증하려는 노력이 장기적으로 실력을 키우는 핵심이라는 점에 팀원 모두 공감했다.</p> <p>네 번째로 '기초 실력을 어떻게 쌓아야 하는가?'를 주제로, SI에게 에러를 바로 넘기기보다 스스로 먼저 해결해보려는 시도, 타인의 코드를 분석하고 직접 짜보는 경험, 그리고 SI가 생성한 코드에 대해 왜 이런 방식으로 구현되었는지 스스로 이해하려는 노력이 중요하다는 이야기를 나눴다. 당장은 시간이 더 걸리고 비효율적으로 느껴질 수 있지만, 이러한 과정이 쌓여야만 장기적으로 실력을 키울 수 있다는 데 의견을 모았다.</p>						
활동 결과						
- 책 소감						

박*진: AI가 코드를 대신 써주는 시대에 개발자란 무엇인가를 진지하게 고민하게 되었다. 단순히 코드를 타이핑하는 사람이 아니라, 결과물을 이해하고 끝까지 책임질 수 있는 사람이 진정한 개발자라는 생각이 들었다. 특히 기초 없이 AI를 사용하면 결과를 검증할 기준 자체가 없어진다는 점이 가장 인상 깊었고, 기초 실력의 중요성을 다시 한번 느꼈다. AI 사용 공개 범위에 대한 논의에서는 프롬프트가 곧 나의 사고 과정이라는 시각이 새로웠다. 이 책을 통해 AI를 두려워하기보다 내 기초와 판단력을 바탕으로 AI를 압도적으로 활용하는 개발자가 되고 싶다는 목표가 생겼다.

이*운: 이 책은 AI, 바이브 코딩 시대에 개발자가 어떻게 생존해야 하는지 구체적으로 가이드를 제시해준다. 특히 컴퓨터공학과 학생이자 주니어 개발자로서 지금 개발을 공부하는게 맞는건지, AI를 어떤 식으로 활용해야 할지 답이 나오지 않아 혼란스러웠는데, 주니어 개발자 파트를 읽고 앞으로의 공부 방향이 정리되었다. 지금까지는 AI가 만들어준 코드를 제대로 이해하지 못한채로 넘어가기도 했고, 에러가 나면 직접 알아보기보다 바로 AI한테 고쳐달라고 에러 코드를 복붙해주시기도 했었다. 하지만 앞으로는 AI가 왜 이런 코드를 작성했는지 생각해보고 에러가 났을 때 AI한테 바로 넘기기도 내가 직접 해결해보려 노력할 것이다. 귀찮고 오래 걸리겠지만 그 과정에서 배우는 게 있을 것 같다.

김*비: 가장 인상 깊었던 점은 개발자가 사라지는 것이 아니라 역할이 변화하고 있다는 부분이었다. 평소에는 AI가 점점 발전하면서 개발자의 필요성이 줄어들 수도 있겠다고 막연하게 생각했는데, 이 책에서는 오히려 그런 변화 속에서 개발자의 사고력과 문제 해결 능력이 더 중요해진다고 말하는 점이 새롭게 다가왔다.

특히 '바이브 코딩'이라는 개념을 통해, 단순히 감에 의존하거나 AI가 만들어준 코드를 그대로 사용하는 방식은 장기적으로 개발자로서 성장하는 데 한계가 있다는 점이 인상 깊었다. 평소 과제나 프로젝트를 하면서 AI의 도움을 자주 받는 편인데, 그 과정에서 코드의 원리를 깊게 이해하지 않고 넘어갔던 경험이 있어서 더 공감됐다.

황*민: 바이브 코딩 너머 개발자 생존법을 통해 바이브 코딩 시대의 개발자는 어떻게 살아가야 할까에 대해 생각해볼 계기가 되었고, AI를 단순히 편의를 위한 도구로만 쓰는 것이 아니라 어떻게 활용해야 하는지에 대해서도 고민해볼게 되었다. 특히 개발자의 역할이 줄어드는 것보다 오히려 더 높은 수준의 판단과 이해를 요구받는 방향으로 변화하고 있다는 점이 인상적이었다. AI가 코드를 대신 작성해줄 수는 있지만, 그 코드의 적절성과 방향성을 결정하는 것은 결국 개발자라는 점에서 개발자의 본질적인 역할은 여전히 중요하다고 느꼈다. 또한 앞으로는 AI가 만들어준 결과를 그대로 사용하는 데서 그치지 않고, 왜 그런 방식으로 구현되었는지 스스로 이해하려는 노력이 필요하다고 생각했다. 당장은 시간이 더 걸리고 비효율적으로 느껴질 수 있지만, 이런 과정이 쌓여야만 장기적으로 실력을 키울 수 있을 것 같다. 따라서 AI에 의존하기보다는 보조 수단으로 활용하면서, 스스로 사고하고 문제를 해결하는 능력을 기르는 방향으로 학습해야겠다고 다짐하게 되었다.

- AI 사용 사실, 어디까지 공개해야 할까?

박*진: 과제에서 AI 사용 사실을 밝히려는 요구를 받고, 대화 내용 전체를 mhtml 파일로 제출한 적이 있다. 제출하고 나서는 내 비법을 전부 털린 느낌이였다. 나는 공개 범위가 과했다라고 생각한다.

AI 사용 공개에는 찬성한다. 다만 사용 여부를 밝히는 것으로 충분하다고 생각한다. 프롬프트는 단순한 입력값이 아니다. 어떤 문제를 어떻게 정의하고, 어떤 방식으로 접근할지를 담은 나의 사고 과정이다. 좋은 프롬프트일수록 그 사람의 도메인 지식과 경험이 녹아 있다. 그걸 통째로 공개하라는 요구는 결과물이 아니라 사고 과정 자체를 내놓으라는 말과 다르지 않다.

공개 목적은 투명성이다. AI를 썼다는 사실을 밝히는 것만으로 그 목적은 충분히 달성된다. 결과물에 AI가 개입했다는 사실을 아는 것과, 어떻게 활용했는지를 낱알이 아는 것은 전혀 다른 차원의 요구다. 전자는 신뢰를 위한 것이지만, 후자는 감시에 가깝다.

AI 사용 사실 표기면 충분하다. 투명성의 이름으로 과정까지 요구하는 건 공개가 아니라 검열이다.

이*운: 코드를 제출할 때 AI를 썼다고 밝혀야 하는지에 대해 아직도 기준이 모호하다. 구글 검색으로 찾은 코드는 괜찮고, Copilot이 자동완성해준 코드는 문제가 되는 걸까? 선이 어디에 있는지 솔직히 잘 모르겠다.

그런데 책을 읽으면서 이 고민의 핵심이 사실 공개 여부가 아니라는 생각이 들었다. 진짜 문제는 내가 그 코드를 이해하고 있느냐다. AI가 만들어줬어도 내가 한 줄씩 설명할 수 있다면 그건 내 코드다. 반대로 직접 짰어도 왜 작동하는지 모른다면, 그건 사실 별로 다를 게 없다. 결국 공개의 기준보다 이해의 기준이 먼저여야 한다고 본다. AI 사용 여부보다 이 코드에 대해 책임질 수 있는가가 더 본질적인 질문이다.

김*비: 우선 AI에게 도움을 받았거나 사용했다면, AI를 사용했다는 사실은 명확하게 표시해야 한다고 생각한다. AI를 어

떻게 활용하느냐에 따라 더 효율적이고 뛰어난 개발자나 디자이너가 될 수는 있지만, 그렇다고 해서 결과물이 전부 자신의 실력만으로 만들어진 것은 아니기 때문이다. 그래서 AI 사용 여부는 솔직하게 밝히는 것이 맞다고 생각한다. 다만 프롬프트까지 반드시 공개해야 한다고는 생각하지 않는다. 프롬프트를 만드는 과정도 하나의 실력이고 노하우라고 생각하기 때문이다. 어떤 방식으로 원하는 결과를 끌어내는지도 결국 능력의 일부라고 느꼈다.

황*민: 가장 확실한 것은 AI를 사용했다는 사실은 무조건 밝혀야 한다. 개발분야에서 사용한다 하면 AI를 통해 프론트엔드나 백엔드 등에서 사용했다면 어디 분야에서 사용했는지 정도는 밝혀야 한다. 다만 직접적으로 그 프롬프트를 밝히는 것은 옳지 않다고 생각한다. 프롬프트를 짜는 능력또한 개개인의 능력이기에 프롬프트의 공개하는것은 오픈소스와 같을수도 있다 라고 생각한다.

- 코드를 승인만 하는 개발자도 개발자인가?

박*진: 나는 그렇다고 생각한다. 단, 조건이 있다.

개발자를 개발자답게 만드는 건 코드를 타이핑하는 행위가 아니다. 결과물을 이해하고 거기에 책임을 지는 것이다. AI가 짠 코드라도 그것을 검토하고 수정하며 최종적으로 책임을 지는 사람이라면, 그 사람은 충분히 개발자라고 부를 수 있다. 반대로 이해 없이 그냥 승인 버튼만 누르는 사람은 다르다. 책임질 수 없는 결과물에 도장을 찍는 것은 개발이 아니다. 기준은 AI를 썼느냐가 아니라, 그 결과물을 온전히 자기 것으로 책임질 수 있느냐다.

이 관점에서 프로그래머와 개발자를 나뉘볼 수 있다. 프로그래머가 코드를 직접 작성하는 사람이라면, 개발자는 결과물을 이해하고 책임지는 사람이다. AI 시대에 프로그래머의 역할은 줄어들 수 있다. 하지만 개발자는 사라지지 않는다. 오히려 더 중요해진다.

이*운: 내 생각엔 승인 자체는 개발 행위가 맞다. 하지만 그 승인이 진짜 판단에서 나온 건지, 아니면 그냥 돌아가니까 넘기는 건지에 따라 달라진다. 책에서 주니어 개발자의 실수로 꼽은 것들(보안 고려사항 누락, 자신도 이해 못 한 시스템 구축 등)은 코드를 직접 짜서 생긴 실수가 아니라 제대로 검토하지 않아서 생긴 실수다. 즉 승인의 질이 문제다. 코드를 짜는 사람이 개발자가 아니라 코드를 책임지는 사람이 개발자라고 생각한다.

김*비: 현재 AI 시대가 되면서 개발자라는 직업의 의미 자체도 조금씩 바뀌고 있다고 생각한다. 예전에는 직접 처음부터 끝까지 코드를 작성하는 사람만 개발자라고 생각하는 분위기가 강했지만, 이제는 AI를 활용해서 더 빠르게 결과물을 만들어내는 방식도 하나의 개발 방식이 되고 있는 것 같다.

물론 AI가 대부분의 코드를 생성했다고 해도, 그것을 검토하고 수정하고 방향을 결정하는 과정은 결국 사람이 해야 한다. 그래서 단순히 "AI가 만들었으니 개발자가 아니다"라고 보기는 어렵다고 생각한다. 중요한 것은 얼마나 제대로 이해하고 활용하느냐인 것 같다.

황*민: 나는 개발자라 생각한다. 코드를 승인하기 위해서는 그 코드가 어디 파트에서 사용되고 코드가 적합한지 판단하며 코드의 문제점을 파악하여 승인하지 않고 수정하는 그러한 전 과정자체를 개발으로 보며 그러한 과정을 총괄하고 승인하는 것은 결국 인간이기에 코드를 승인만 하더라도 개발의 과정을 참여하며 이끌어가는 개발자라 생각한다.

- 기초 없이 AI를 쓰면 독이 된다 — 그렇다면 기초란 무엇인가?

박*진: 기초 없이 AI를 쓰면 독이 된다는 주제에서 출발해, "기초"의 의미를 단계적으로 파고들었다. 코드를 이해한다는 것은 단순히 읽는 것을 넘어 의도를 파악하고 버그를 찾을 수 있는 수준이어야 하며, 그러려면 결국 직접 짤 수 있는 능력이 기반이 되어야 한다는 결론에 이르렀다. 알고리즘 문제 풀기는 특정 알고리즘 자체보다 그 과정에서 형성되는 사고방식이 진짜 가치이며, 디테일은 까먹더라도 사고방식의 틀이 남아 있다면 AI로 메우는 것은 기초를 효율적으로 활용하는 것이지 기초가 없는 것이 아니다. 기초 없이 AI를 쓰는 것의 진짜 위험은 결과를 검증할 수 없다는 것이다.

이*운: 책은 기초를 갖춰야 AI를 제대로 이끌 수 있다고 말하는데, 그 기초가 구체적으로 뭔지는 사람마다 다르게 읽힌다. 자료구조, 알고리즘, 메모리 관리 같은 CS 지식을 말하는 건지, 아니면 디버깅하고 트레이드오프를 따지는 사고방식을 말하는 건지. 책은 둘 다라고 하지만, 나는 후자가 더 핵심이라고 본다. 특정 언어나 프레임워크는 바뀌어도, 왜 이 방식이 저 방식보다 나은가를 따지는 능력은 잘 안 바뀐다. AI가 틀린 코드를 냈을 때 이를 알아채는 건 암기된 지식이 아니라 그 사고방식에서 나온다. 그래서 나한테 기초는 특정 지식의 총량이 아니라 코드 앞에서 질문을 멈추지 않는 태도에 가깝다. 책에서 말하는 왜라는 질문을 건너뛰지 말 것이 그걸 잘 요약한다고 생각한다.

김*비: 기초가 부족한 상태에서 AI만 의존하게 되면 결국 스스로 생각하는 힘이 약해질 수 있다고 생각한다. 물론 AI의

도움을 적절하게 받는 것은 학습에도 도움이 되지만, 사람은 원래 더 쉽고 빠른 길을 선택하고 싶어하기 때문에 점점 의존하게 되는 것 같다.

실제로 나도 대학교 1학년 때부터 AI의 도움을 많이 받으면서 공부했는데, 그때 기초를 충분히 다치지 못한 부분들이 지금 와서 어렵게 느껴질 때가 많다. 특히 코드의 원리나 기본 개념을 제대로 이해하지 못한 채 넘어간 부분에서 많이 막혔다. 그래서 결국 가장 중요한 것은 기본기를 단단하게 쌓는 것이라고 생각한다. AI를 활용하는 것은 좋지만, 스스로 이해하고 고민하는 과정 없이 사용하게 되면 오히려 독이 될 수도 있다고 느꼈다.

황*민: 기초없이 AI만 사용한다면 자신이 무엇을 하고 있는지조차 모르고 그저 AI가 짜는 코드를 구경만 할 뿐이다. 내가 생각하는 기초란 코드를 보고 이해하며 개선해야 하는 방향을 아는 것 이라 생각한다 이를 기르기 위해 평소에 타인이 짠 코드를 분석해보고 자신이 직접 코드를 짜보는 것이 중요하다 생각한다

활동 자료

<참고 자료>

"구글 신규 코드 75%는 AI 작성"...코딩 속도는 빨라졌지만 취약점도 늘었다

정재엽 기자 | 2026.04.24



구글이 내부에서 작성하는 신규 코드의 4분의 3을 AI가 생산하는 시대가 열렸다. 개발 속도 혁명은 현실이 됐지만, 그 이면에서 보안 위협이라는 새로운 과제가 수면 위로 올라오고 있다.

순다르 피차이 구글 CEO는 지난 22일(현지시간) 미국 라스베이거스에서 열린 '구글 클라우드 넥스트 2026' 기조연설에서 현재 구글 엔지니어가 검토하고 채택하는 신규 코드 가운데 75%가 AI로 작성된 것이라고 공개했다. 지난해 가을 집계했던 50%에서 반년 만에 크게 뛰어오른 수치다.

피차이 CEO는 이를 '코딩 패러다임의 전환'으로 규정하며, 엔지니어의 핵심 역할이 코드 생성에서 AI 산출물의 검토·통합으로 이동하고 있다고 강조했다.

<활동 사진>

북클럽 팀명	NextDev					
일시/장소	일시	2026.05.12			장소	도서관 3층
참여 인원	학과	빅데이터	학번	2024****	이름	박*진
	학과	컨텐츠IT	학번	2024****	이름	이*운
	학과	스마트IOT	학번	2024****	이름	김*비
	학과	소프트웨어 AI 계열	학번	2026****	이름	황*민
	참석인원(명)	4 / 4			참가자 대표	박*진 (서명)
도서	도서명	먼저 온 미래			저자	장강명
	출판사	동아시아	출판연도	2025	번역자	
오늘의 북클럽 주제	개발자를 넘어, AI 시대					
활동 내용 & 방법						
<p>책을 읽은 후, 인상 깊었던 내용을 나누기 전에 먼저 각자 책 소감을 자유롭게 이야기하는 시간을 가졌다. 알파고 사건이 당시 사람들에게 얼마나 큰 충격이었는지, AI가 단순한 기술을 넘어 다양한 직업군의 삶과 생각에 어떤 영향을 미치는지, 그리고 기술의 가능성과 한계를 동시에 바라보는 균형 잡힌 시각의 필요성 등 각자가 인상 깊게 읽은 부분을 공유하며, 기술이 빠르게 변화하는 시대를 어떻게 살아갈 것인지에 대한 생각을 나눴다. 이후 사전에 준비한 핵심 주제를 바탕으로 본격적인 토론을 진행했다. 주요 토론 주제는 다음과 같았다.</p> <p>첫 번째로 'AI가 만든 결과물과 인간이 만든 결과물은 다른 가치를 지니는가?'를 논의하며, 두 결과물이 서로 다른 가치를 지닌다는 데 팀원 모두 공감했다. AI 결과물은 빠른 생산성과 효율성에서, 인간의 결과물은 경험과 감정, 개성에서 각각 고유한 가치를 가진다는 의견이 중심을 이루었다. 다만 AI가 만든 결과물이라도 그것을 기획한 사람의 의도와 문제의식이 담겨 있다면 충분히 인간적인 가치를 지닐 수 있다는 점에서, 결과물의 가치는 도구가 아닌 그 뒤에 담긴 사람의 생각과 감각으로 판단해야 한다는 이야기를 나눴다.</p> <p>두 번째로 'AI가 정답을 알려주면 탐구의 즐거움은 사라지는가?'를 주제로 토론했다. 팀원 전원이 사라지지 않는다는 입장을 취했다. 계산기가 생겼다고 수학적 사고를 배울 필요가 없어지지 않은 것처럼, AI가 답을 준다고 해서 생각하는 과정이 사라지는 않는다는 의견이 모였다. 오히려 AI 덕분에 평소에 도전하지 못했던 분야까지 탐구의 범위가 넓어지고, AI에게 어떻게 질문을 던질지 고민하는 것 자체가 이미 탐구의 시작이라는 이야기를 나눴다.</p> <p>세 번째로 '기술은 문제를 만드는가, 해결하는가?'를 논의하며, 기술은 문제를 해결하는 동시에 새로운 문제를 만들어내는 순환 과정이라는 데 의견이 모였다. 자동차가 교통사고라는 문제를 낳았지만 안전벨트와 자율주행 기술이 그것을 해결해나가듯, 기술과 문제는 서로를 추동하며 함께 진화한다는 점에서 새로운 문제의 등장을 부정적으로만 볼 필요는 없다는 결론에 이르렀다. 다만 기술의 혜택이 모두에게 균등하게 돌아가지 않는다는 점은 여전히 열린 질문으로 남겼다.</p> <p>네 번째로 'VR/AR 환경이 일상이 되는 것은 가능할까?'를 주제로, 가능성 자체는 충분히 열려 있다는 데 팀원 모두 동의했다. 코로나 시기 온라인 수업과 재택근무를 통해 가상 환경이 일상에 침투하는 경험은 이미 시작되었다는 의견도 나왔다. 다만 기기의 무게, 멀미 문제 등 현실적인 장벽이 남아 있으며, 이러한 기술적 한계가 해결되는 속도가 VR/AR의 일상화 시점을 결정할 것이라는 이야기를 나눴다. 나아가 공통 현실이 사라질 수 있다는 점은 기술의 가능성과는 별개로 깊이 생각해야 할 문제라는 의견도 제기되었다.</p> <p>다섯 번째로 '안정적인 AI의 결과물을 이기기 위해서는 어떻게 해야 할까?'를 논의하며, AI를 이기려는 프레임보다 AI가 대체할 수 없는 영역을 키우는 것이 더 현실적인 방향이라는 데 의견이 모였다. AI는 패턴 인식과 반복, 최적화에 강하지만, 문제를 발견하고 맥락 속에서 의미를 찾는 능력은 인간 고유의 영역이라는 점에서 자신만의 경험과 개성을 갈고</p>						

뒀는 것이 중요하다는 이야기를 나눴다. 또한 기초 없이 AI를 사용하면 결과물을 평가하거나 개선할 기준 자체가 없어지기 때문에, 탄탄한 기초를 바탕으로 AI를 주도적으로 활용하는 것이 결국 압도적인 결과물을 만드는 길이라는 결론에 이르렀다.

활동 결과

- 책 소감

박*진: 기술의 발전이 단순히 문제를 해결하는 것을 넘어 새로운 문제를 만들어내는 순환 과정임을 깨달았다. AI가 정답을 빠르게 제시해준다고 해서 탐구의 즐거움이 사라지는 것이 아니라, 오히려 더 많은 질문을 던질 수 있는 발판이 된다는 시각이 신선하게 다가왔다. AI가 만든 결과물도 그것을 기획한 사람의 의도와 문제의식이 담겨 있다면 충분히 인간적인 가치를 지닐 수 있다는 점도 인상 깊었다. VR/AR이 일상이 되기 위한 현실적 장벽처럼, 기술의 가능성과 한계를 동시에 바라보는 균형 잡힌 시각을 갖게 되었다. 미래는 이미 와 있고, 그 안에서 어떤 질문을 던지며 살아갈지가 결국 기술 시대를 살아가는 인간의 몫이라는 것을 느꼈다.

이*윤: 먼저 온 미래를 읽으며 AI 시대에 어떤 직업군이 타격을 받고 세상이 어떻게 바뀌어가는지 생각해볼 수 있었다. 기술이 발전한다는 건 인간의 일이 줄고 삶이 편리해진다는 의미라고 믿어왔는데, 오히려 일자리가 줄어들어 사람들이 궁지에 몰리는 현실을 보며 많은 생각이 들었다. 우리 삶은 앞으로 어떻게 될까? 산업혁명 때처럼 새로운 일자리가 창출되어 사람들이 또 다른 길을 찾게 될까, 아니면 레디 플레이어 원처럼 디스토피아 속에서 가상현실에 의존하며 살아가게 될까.

김*비: IT계에서 알파고 사건이 엄청난 사건이었다는 것은 알고 있었지만, 당시에는 어리기도 했고 뉴스나 영상으로 간단하게만 접했어서 크게 관심을 가지진 않았었다. 그런데 이 책을 읽으면서 알파고 사건이 당시 사람들에게 얼마나 큰 충격이었는지, 그리고 AI 기술이 얼마나 빠르게 발전했는지를 다시 느끼게 되었다.

바둑은 단순히 머리만 쓰는 게임이 아니라 예술, 창의성, 감각의 영역이라고 생각했기 때문에 그런 바둑에서 인간이 AI에게 진다는 사실 자체가 사람들에게 더 큰 충격으로 다가왔던 것 같다. 실제로 바둑계에서는 충격을 받고 은퇴한 사람들도 있었지만, 반대로 AI를 적극적으로 도입해서 공부와 분석에 활용하면서 전 세계 바둑 수준 자체가 더 올라갔다는 점이 아이러니하면서도 굉장히 인상 깊었다. 인간의 영역이라고 생각했던 분야도 AI와 공존하면서 또 다른 방향으로 발전할 수 있다는 점이 신기하게 느껴졌다.

황*민: 먼저 온 미래를 통하여 개발자의 입장이 아닌 다른 직업군의 입장에서의 AI에 대한 관점을 볼 수 있었기에 좋았던 것 같다. 그동안 읽었던 책들은 현직 실무자나 관련 분야 사람들의 시선에서 기술적인 설명에 집중된 경우가 많았지만, 이 책은 소설 작가의 관점, 바둑 기사들의 관점 등 여러 직업군의 시선을 통해 AI를 바라볼 수 있었다는 점에서 차별점을 느꼈다. 특히 같은 AI 기술이라도 이를 받아들이는 방식이 직업에 따라 전혀 다르게 나타난다는 점이 인상 깊었다. 누군가에게는 분석과 효율을 높여주는 도구로 작용하지만, 다른 누군가에게는 자신의 영역이 위협받는 존재로 느껴질 수도 있다는 점에서 AI가 단순한 기술 이상의 의미를 가진다는 것을 깨닫게 되었다. 이처럼 다양한 입장을 통해 AI를 바라보니, 기술 자체에 대한 이해를 넘어서 그것이 실제 사람들의 삶과 생각에 어떤 영향을 미치는지까지 생각해볼 수 있었고, 이러한 점이 이 책을 더 의미 있게 느끼게 해주었다.

- AI가 만든 결과물과 인간이 만든 결과물은 다른 가치를 지니는가?

박*진: 요즘 사회에서 결과물의 가치는 그 퀄리티 자체보다 그 뒤에 담긴 인간의 경험과 감정에서 비롯된다. AI가 생성한 결과물이라 하더라도 그것을 만들어내기 위해 어떤 의도로 프롬프트를 설계했는지, 어떤 문제의식에서 출발했는지에 따라 가치는 달라진다. 결국 AI는 도구이고, 그 도구를 통해 얼마나 인간의 본질이 드러나느냐가 결과물의 진짜 가치를 결정한다. 붓으로 그린 그림과 디지털로 그린 그림의 가치를 단순히 도구로 나누지 않듯, AI 결과물도 그것을 만든 사람의 생각과 감각이 얼마나 잘 녹아있느냐로 평가받아야 한다.

이*윤: 우리가 예술이나 글에서 감동받는 건 결과물 자체보다 그 뒤에 있는 맥락과 과정 때문일 때가 많다. 가치는 결과물에만 있는 게 아니라 그것이 만들어진 과정에도 있다. 그러나 AI의 결과물을 단순히 열등하다고 볼 수만은 없다. AI는 수많은 데이터를 참조해 인간이 쉽게 떠올리지 못할 방향으로 조합해내기 때문에, 그 자체로 새로운 자극과 즐거움을 줄 수 있다. 결국 AI와 인간의 결과물은 우열의 문제가 아니라 다른 종류의 가치를 지닌다고 보는 게 맞을 것 같다.

김*비: 사전적 의미에서 가치는 '좋은 것', '값', '값어치가 있는 것'을 의미한다. 가치란 시대나 상황에 따라 변할 수 있

는 상대적인 개념이라고 생각하기 때문에 AI와 인간이 만든 결과물은 서로 다른 가치를 가진다고 생각한다. 그렇다고 해서 어느 쪽의 가치가 더 높거나 낮다는 의미는 아니다. 단지 가치의 방향이 다르다고 생각한다.

AI 결과물의 가치는 빠른 생산성, 효율성, 대량 생성 가능, 일정한 품질 같은 부분에 있다고 생각한다. 짧은 시간 안에 원하는 결과물을 만들어낼 수 있고 누구나 쉽게 활용할 수 있다는 점에서 실용적인 가치가 크다. 반대로 인간의 결과물은 경험, 감정, 의도, 개성, 과정 같은 부분에서 가치가 있다고 생각한다. 사람들은 단순히 결과만 보는 것이 아니라 그 사람이 어떤 생각과 감정을 담아서 만들었는지에도 의미를 느끼기 때문이다. 그래서 AI는 실용성과 효율의 가치가 크고, 인간은 감정과 의미의 가치가 더 크다고 생각한다.

황*민: AI의 결과물과 인간의 결과물은 둘다 가치를 지니지만 둘의 가치는 다르다 생각한다. AI의 결과물또한 그 만의 가치가 있다. 하지만 공장제품과 수제 제품의 차이가 있듯 AI의 결과물과 인간의 결과물또한 서로 다른 가치를 지닐 것이라고 생각한다.

- AI가 '정답'을 알려주면 탐구의 즐거움은 사라지는가?

박*진: AI가 정답을 빠르게 제시한다고 해서 탐구의 즐거움이 사라지는 것은 아니다. 오히려 답을 찾는 데 드는 시간이 줄어드는 만큼, 그 시간에 더 많은 질문을 던질 수 있게 된다. 계산기가 생겼다고 해서 수학적 사고를 배울 필요가 없어지지 않은 것처럼, AI가 답을 준다고 해서 생각하는 과정이 사라지지 않는다. 오히려 AI에게 어떻게 질문을 던질지 고민하는 것 자체가 이미 탐구의 시작이다. 탐구의 형태가 '하나의 답을 깊이 파고드는 것'에서 '답을 발판 삼아 다음 질문으로 계속 나아가는 것'으로 변화한 것일 뿐, 방식이 쉬워질수록 탐구의 빈도는 오히려 늘어난다.

이*운: 계산기가 발명되었다고 해서 학교에서 수학을 가르치지 않는 건 아니다. 정답을 맞히는 것도 중요하지만, 그 과정에서 사고하는 능력을 기르는 것이 더 본질적인 목표이기 때문이다. 또한 모든 탐구가 정답을 찾기 위한 것은 아니다. 탐구 자체에서 즐거움을 느끼기 위해 어떤 일을 하는 경우도 많다. AI가 정답을 대신 내준다고 해서 탐구의 즐거움이 사라지는 게 아니라, 어떤 탐구를 선택하느냐의 문제로 바뀌는 것이다.

김*비: 나는 그렇지 않다고 생각한다. AI라고 해서 모든 것을 알고 있는 것은 아니기 때문이다. 오히려 AI를 사용하면 서도 AI가 모르는 부분이나 부족한 부분을 직접 탐구하고 알아가는 과정에서 또 다른 재미를 느낄 수 있다고 생각한다. 그리고 스스로 고민해서 답을 찾아냈을 때 AI도 생각하지 못한 부분을 발견했다는 느낌이 들어 더 큰 성취감과 즐거움을 느낄 수도 있다고 생각한다.

황*민: 아니라고 생각한다 오히려 AI의 등장으로 정보를 얻는 것이 쉬워지며 평소에 도전하지 못하던 분야에 대해서도 탐구 해 볼 수 있었던 것 같다. 실제로 AI가 상용화 된 후 평소에 도전해 보지 못했던 아두이노를 AI를 이용하여 탐구 해본 경험이 있기에 AI의 등장은 탐구의 즐거움을 사라지게 한 것이 아닌 탐구의 즐거움을 촉진 시켰다 생각한다.

- 기술은 문제를 만드는가, 해결하는가?

박*진: 기술은 문제를 해결하는 동시에 새로운 문제를 만들어낸다. 하지만 이것이 기술 발전을 부정적으로 볼 이유가 되지는 않는다. 새로운 문제는 새로운 기술로 다시 해결할 수 있기 때문이다. 자동차가 교통사고라는 문제를 낳았지만 안전벨트, 에어백, 자율주행 기술이 그 문제를 계속해서 줄여나가듯, 기술과 문제는 서로를 추동하며 함께 진화한다. 중요한 것은 기술이 문제를 만드냐 해결하느냐의 이분법이 아니라, 새로운 문제가 생겼을 때 그것을 다음 기술 발전의 방향으로 삼을 수 있느냐는 태도다.

이*운: 기술은 문제를 해결하는 동시에 문제의 형태를 바꾼다. 식기세척기가 설거지 문제를 해결해줬지만, 생긴 여유 시간을 사람들은 또 다른 경쟁에 쏟는다. 그렇다면 진짜 질문은 새로 생긴 문제가 이전 문제보다 나은가이다. AI로 인해 많은 것이 자동화되고 발전하고 있지만, 일자리를 잃은 사람들은 어디로 가야 하는가. 기술의 발전이 모두에게 균등한 혜택으로 돌아오지 않는다는 점에서 이 질문은 아직 열려 있다.

김*비: 기술은 문제를 해결하기도 하지만 동시에 새로운 문제를 만든다고 생각한다. 예전에는 오래 걸리던 일을 기술 덕분에 빠르게 해결할 수 있게 되었지만, 그만큼 사람들의 기준과 경쟁 수준도 더 높아진 것 같다. AI 역시 인간의 일을 줄여주는 것처럼 보이지만 결국 더 빠른 생산성과 더 높은 능력을 요구하게 만드는 느낌도 들었다. 그래서 기술은 단순히 문제를 없애는 것이 아니라 인간 사회의 문제를 다른 형태로 바꾸는 것 같다고 느꼈다.

황*민: 기술의 등장은 문제를 해결하기보단 새로운 문제의 시작이하 생각한다 다만 새로운 기술이 등장하고 그로인한 문제가 생겨나고 그 문제를 해결하려 새로운 기술이 등장하는 과정속에서 우리 사회가 더욱 발전하고 살기좋은 사회가

되기에 새로운 문제의 시작이지만 부정적으로 볼 필요는 없다고 생각한다.

- vr/ar 환경이 일상이 되는 것은 가능할까?

박*진: 이미 많은 사람들이 게임이나 온라인 커뮤니티 안에서 현실 못지않은 삶을 살아가고 있다고 생각하며, 코로나 시절에 메타버스를 통해 가상 환경이 일상에 침투하는 경험은 이미 시작되었다는 의견에 동의한다. 가능성 자체는 충분히 열려 있다. 다만 VR/AR이 진정한 일상이 되기 위해 넘어야 할 가장 현실적인 장벽은 멀미 문제다. 아무리 몰입감이 높아도 신체가 거부하면 일상적으로 사용하기 어렵다. 이 생리적 한계를 얼마나 빠르게 해결하느냐가 VR/AR이 대중의 일상으로 자리잡는 속도를 결정할 것이다.

이*운: 스마트폰이 없던 시절엔 하루 10시간씩 화면을 들여다본다는 게 비현실적으로 들렸을 것이다. 지금의 VR 기기는 조작이 제한적이고 멀미라는 단점도 있지만, 기술의 발전 속도를 생각하면 이 한계가 오래가지 않을 것이다. 문제는 가능하냐가 아니라, 그것이 일상이 됐을 때 우리가 공유하던 현실이 어떻게 달라지느냐다. 장강명의 말처럼, 공통 현실이 사라지는 것은 기술의 가능성과는 별개로 생각해야 할 문제다.

김*비: 가능은 하다고 생각하지만 아직 해결해야 할 문제가 많다고 생각한다. 현재는 기기가 무겁거나 오래 사용하면 멀미가 나는 문제도 있고, 기술적인 한계도 아직 존재한다고 생각한다. 그래서 완전히 일상화되기까지는 시간이 더 필요할 것 같고, 아직은 미래의 기술에 더 가깝다고 느껴진다. 하지만 이런 문제들이 점점 해결된다면, VR/AR 환경이 일상이 되는 것도 충분히 가능하다고 생각한다.

황*민: 코로나 19 팬데믹시기에 방역을 위해 학교에선 온라인 수업을 진행하고 회사에선 재택근무를 할때 우리는 이미 메타버스를 체험했다. 그때는 물론 메타버스라 말하기 좀 빈약했지만 vr,ar기술이 더 발전한다면 vr, ar 환경이 일상이 되는 진정한 의미의 메타버스가 가능할 것 이라 생각한다.

- 안정적인 ai의 결과물을 이기기 위해서는 어떻게 해야할까?

박*진: AI는 평균적으로 안정적인 결과물을 빠르게 만들어낸다. 그 평균을 넘어서려면 결국 사람의 손과 머리가 직접 개입해야 한다. 그리고 그 개입이 의미를 가지려면 해당 분야의 탄탄한 기초 지식이 전제되어야 한다. 기초 없이 AI를 사용하면 결과물을 평가하거나 개선할 기준 자체가 없기 때문이다. AI가 만들어낸 결과물의 빈틈을 발견하고, 거기에 인간만의 맥락과 감각을 더할 수 있는 사람이 결국 AI와 경쟁하는 것이 아니라 AI를 활용해 압도적인 결과물을 만드는 사람이 될 것이다.

이*운: AI가 잘하는 건 패턴 인식, 반복, 최적화다. 반면 인간은 문제를 발견하고, 다양한 맥락 속에서 그 의미를 생각하며, 실패 안에서도 방향을 찾아간다. AI를 이기려는 프레임보다, AI가 대체할 수 없는 영역을 키우는 것이 더 현실적인 답일 수 있다. 결국 AI 시대에 필요한 역량은 더 빠른 실행이 아니라 더 좋은 질문을 던지는 능력이다.

김*비: 나는 결국 자신의 개성을 더 발전시키는 것이 중요하다고 생각한다. AI는 결국 학습된 데이터를 기반으로 결과를 만들어내는 것이기 때문에 완전히 새로운 개인의 경험이나 감정까지 그대로 따라오기는 어렵다고 생각한다. 그래서 사람은 자신의 경험과 개성을 계속 갈고닦아서 AI도 쉽게 따라 할 수 없는 자신만의 결과물을 만드는 방향으로 가야 한다고 생각한다.

황*민: 안정적으로 뽑아내는 AI의 코드또한 인간의 결정이 없다면 만들어 질 수 없다. 그래도 AI코드를 사용하면서 기초 없이 그저 승인만 하며 사용한다면 그 결과물은 도태될 것이다. 그러므로 AI 의 코드를 이기기 위해선 기초를 잘 단련하는 것이 중요하다 생각한다.

활동 자료

북클럽 팀명	NextDev					
일시/장소	일시	2026.05.13		장소	CLC 3층	
참여 인원	학과	빅데이터	학번	2024****	이름	박*진
	학과	컨텐츠IT	학번	2024****	이름	이*운
	학과	스마트IoT	학번	2024****	이름	김*비
	학과	소프트웨어 AI 계열	학번	2026****	이름	황*민
	참석인원(명)	4 / 4		참가자 대표	박혜진 (서명)	
도서	도서명	-			저자	-
	출판사	-	출판연도	-	번역자	-
오늘의 북클럽 주제	글로벌 현직자 특강 : AI가 일하는 시대, 인간은 무엇을 하는가					
활동 내용 & 방법						
<p>북클럽 주제인 '기술 변화 속 인간 개발자의 정체성과 가치 탐구'와 관련된 심화 자료로 'AI가 일하고, 인간은 무엇을 하는가?'라는 주제의 Microsoft AI Skilling 특강을 함께 청취하였다. 특강은 AI가 업무를 대체해가는 시대에 인간에게 남는 고유한 역할과 가치가 무엇인지를 다루고 있어, 개발자의 정체성과 가치를 탐구하는 북클럽 주제와 깊이 맞닿아 있다고 판단하여 심화 자료로 활용하였다.</p> <p>특강 청취 후 팀원 각자가 가장 인상 깊었던 내용과 느낀 점을 자유롭게 나누며 소감을 공유하였다.</p>						
활동 결과						
<p>- 특강 소감</p> <p>박*진: 특강과 토론을 통해 AI 시대에 개발자에게 요구되는 핵심 역량이 "코드를 짜는 능력"에서 "의도를 설계하고 전달하는 능력"으로 변화하고 있음을 깨달았다.</p> <p>직접 파인만 학습 프로그램을 만들었던 경험을 돌아보면, AI가 실행을 담당했지만 나의 학습 스타일에 맞는 프롬프트를 설계하고 전달하는 것은 온전히 나의 역할이었다. 처음에는 단순히 공부 효율을 높이기 위해 시작했지만, 어느 순간 프로그램을 만드는 과정 자체에 몰입하고 즐기고 있는 나를 발견했다. 이것이 특강에서 말한 "인간이 가장 잘하는 것은 노는 것"과 연결되는 순간이었다. 효율을 위해 시작했지만 즐기면서 몰입했던 그 경험이야말로, AI가 흉내 낼 수 없는 인간 고유의 방식이라는 것을 깨달았다.</p> <p>또한 특강에서 앞으로 회사에서는 개인 에이전트를 만드는 것이 필수가 될 것이라는 말이 인상 깊었다. AI를 단순히 도구로 사용하는 것을 넘어, 나만의 에이전트를 설계하고 운영하는 에이전트 보스의 역할이 개발자에게도 요구될 것이라는 점에서, 앞으로 코드를 잘 짜는 능력만큼이나 자신의 의도를 명확히 정리하고 설계하여 전달하는 능력을 길러야겠다고 생각했다.</p> <p>이 북클럽에서 책을 읽고 생각을 정리하고 토론에서 언어로 전달하는 과정 자체가, 결국 그 능력을 기르는 훈련이 되고 있다는 것도 새삼 느꼈다.</p>						

이*윤: 강의에서 가장 인상 깊었던 것은 프론티어 기업으로 가는 3단계였다. AI 어시스턴트에서 에이전트로, 그리고 팀 단위 에이전트로. 앞으로 AI를 얼마나 깊이 업무에 녹여내느냐에 따라 기업 간 격차가 크게 벌어질 것이다. AI 활용이 이제 선택이 아닌 생존의 문제임을 실감하게 해준다.

그렇다면 나는 에이전트 보스가 되기 위해 무엇을 준비해야 할까. 강사는 프롬프트를 나의 의도를 에이전트가 이해할 수 있는 언어로 번역하는 행위라고 표현했다. 처음엔 프롬프트를 잘 쓰는 것이 일종의 기술처럼 느껴졌지만, 생각해보면 결국 도메인 지식이 탄탄한 사람이 더 좋은 프롬프트를 쓸 수 있다. 무엇을 물어야 하는지, 어떤 맥락을 줘야 하는지를 아는 것 자체가 이미 실력이기 때문이다.

인간이 가장 잘하는 것은 노는 것이라는 마지막 메시지도 오래 남았다. AI가 최적화를 담당하는 시대일수록, 왜 이 일을 해야 하는지를 묻고 새로운 방향을 상상하는 힘은 여전히 사람의 몫이다. 그 힘을 키우기 위해서라도 지금 내가 쌓아가는 기초가 더 중요하다는 생각이 들었다.

김*비: AI가 발전할수록 인간의 역할이 줄어들 것이라고 생각했는데, 오히려 인간의 역할 자체가 다른 방향으로 변화하고 있다는 점이 인상 깊었다. 또 기억에 남았던 것은 '에이전트 보스'라는 개념이었다. 앞으로는 단순히 개발 능력만 중요한 것이 아니라, AI에게 원하는 결과를 정확하게 설명하고 목표를 전달하는 능력도 굉장히 중요해질 것 같다고 느꼈다. 평소 AI를 과제나 자료 정리 정도에만 활용했는데, 실제 기업에서는 업무 전체를 AI와 연결해서 사용하고 있다는 점도 신기했다.

황*민: 이번 특강에서 가장 인상 깊었던 것은 AI가 단순한 도구를 넘어 실제 업무 전반에 깊이 활용되고 있다는 점이었다. 특히 AI 적용 점수(AI Adoption Score)라는 개념을 통해, 단순히 AI를 '사용한다'는 것과 실제 업무에 '잘 활용한다'는 것은 전혀 다른 문제라는 것을 알게 되었다. 앞으로는 AI를 얼마나 자연스럽게 업무에 녹여내느냐에 따라 개인과 기업의 경쟁력이 크게 달라질 것이라는 생각이 들었다.

또한 Microsoft 365 Copilot과 같은 도구를 통해 이메일, 문서, 일정 등 다양한 업무가 하나로 연결되어 자동화된다는 점도 인상적이었다. 평소에는 AI를 과제나 자료 정리 정도로만 활용했는데, 실제 기업에서는 업무 흐름 전체를 AI와 연결해 활용하고 있다는 점이 새롭게 다가왔다.

마지막으로 '인간이 가장 잘하는 것은 노는 것'이라는 말이 기억에 남았다. 이는 단순한 휴식이 아니라, 정해진 틀에 얽매이지 않고 자유롭게 생각하고 새로운 아이디어를 만들어내는 능력을 의미한다고 느꼈다. AI가 효율과 최적화를 담당하는 시대일수록, 이러한 창의적인 사고가 더 중요해질 것이라고 생각했다.

이번 특강을 통해 AI를 단순한 기술로 보는 것이 아니라, 앞으로의 학습 방향과 역할에 대해 고민해보는 계기가 되었다.

활동 자료

--