

협업을 위한 소프트웨어 공학

이창공

2023년 12월 7일

Abstract

협업은 현대의 연구에서 필수불가결한 요소로 자리매김하고 있다. 전통적인 의미의 소프트웨어 공학은 체계적인 방법론과 원리를 적용하여 소프트웨어를 개발하는 기술을 지칭한다. 소프트웨어 공학의 핵심 요점을 이해한다면, 효율적인 협업을 통해 빠르고 훌륭한 연구 결과를 얻을 수 있을 것이다.

협업은 현대의 연구에서 필수불가결한 요소로 자리매김하고 있다. 이는 효율적인 협업이 곧 기술과 공학의 결합으로 이루어지기 때문이다. 특히 프로그램을 개발하는 과정에서는 협업이 더욱 중요하다. 작업의 복잡성 때문에 항상 백지장도 맞들면 나은 상황이 아니기 때문이다. 심지어 협업이 잘 이루어지지 않으면, 개발자들은 서로의 코드를 이해하지 못하고, 불필요한 코드를 작성하게 되고, 결국 혼자 하느니만 못한 상황이 된다. 반대로 효율적인 협업은 인원수 이상의 효과를 낼 수도 있을 것이다.

전통적인 의미의 소프트웨어 공학은 체계적인 방법론과 원리를 적용하여 소프트웨어를 개발하는 기술을 지칭한다. 소프트웨어 공학은 오래전부터 수많은 변천사를 겪으며 확장하여 현재의 모습을 갖추게 되었다. 하지만, 1960년대에 나토에 의해 시작된 소프트웨어 공학의 학술적인 기본 원리는 여전히 유효하다. 소프트웨어 공학의 궁극적인 목표는 개발 과정을 계획하고 일정을 맞추어 최소의 비용으로 품질 좋은 소프트웨어를 생산하는 것, 그리고 이러한 활동의 지침을 제공하는 것이다. 이러한 지침에는 단계적 프로세스, 품질 보증, 프로젝트 관리 등 다양한 요소가 있지만, 이 중 연구에 중점을 두는 관점에서는 계획, 자원 관리, 수행 및 진행 관찰 등의 방법론들을 적용할 수 있다.

소프트웨어 공학의 핵심 요점을 이해한다면, 효율적인 협업을 통해 빠르고 훌륭한 연구 결과를 얻을 수 있을 것이다. 많은 협업이 그러하지만 높은 단계의 계획이 뚜렷하더라도 낮은 단계의 계획은 끊임없이 개선될 수 있기 때문에 실제 구현 단계에서 계획을 수정하는 것은 자연스러운 일이다. 이러한 수정은 결국 계획의 자원 관리와 수행 및 진행 상황 관찰의 모든 과정을 통해 이루어진다. 따라서, 개발 과정에서의 소통은 단순히 높은 단계의 아이디어 공유뿐만 아니라 세부 계획의 수정을 통해 이루어질 필요가 있다.