

제7장 경험주의: 현실을 자각하자

- 경험주의 과학 철학에서 실험에 발견한 증거를 강조하는 것으로 정의

7.1 꿈은 높게,그러나 발은 땅에

- 다른 측면은 상용 서비스 중인 시스템이 항상 우리를 놀라게 만들며, 또 그래야 만 마땅하다는 것이다

이는 다른 분야의 과학과 공학에서 배울 수 있는 중요한 교훈이다. 문제 해결을 위한 과학적이고 합리적인 접근 방식에서 매우 중요한 측면 중 하나는 회의주의라는 개념이다. 현실에 대한 증거와 관찰을 바탕으로 의사결정을 내리는 경험주의는 합리적인 발전을 이루기 위해 필수적인 요소다.

7.2 실험과 경험은 분리해야 한다

- 우리는 실험의 일부로 수집한 정보를 사용해 의사결정을 내림으로써 경험주의자가 될 수 있다 우리는 아이디어의 결과를 덜 형식적으로 관찰함으로써 경험주의자가 될 수 있다.

7.3 저 이 버그 알아요

- 저자는 금융거래소에서 a부터z까지 구축하는 경험을 했다. 이 시기 동안 저자는 소프트웨어 개발 접근 방식에 공학적인 사고와 규율을 진지하게 받아들이기 시작했다

버그가 발생했었는데 겉으로 보이는 '메시징 문제'는 원인이 아니라 증상일 뿐이었다 메시징과는 무관한 새로운 코드에서 스테딩 버그가 발생한 것이었다. 메시징 문제라고 선불리 결론을 내리지 않았다면 호들갑을 떨지 않고서도 5분 안에 해결할 수 있는 명백하고 간단한 문제였다

이런 행위만이 우리가 상상했던 문제가 아니라 우리가 당면했던 문제를 해결하기에 충분한 더 많은 사실을 수집하도록 촉진했다

7.4 자기만의 우리의 적

- 경험주의자가 되기 위해서는 현실에서 수집한 신호를 어떻게 조합하고 실험을 통해 테스트 할 수 있는 이론으로 조립하는지에 대해 체계화할 필요가 있다

우리가 보는 것의 대부분은 뇌가 만들어 낸 추측이다 우리가 생존을 위해 싸우고 있던 시절에 시간을 들여 시야를 정밀하게 분석했다면 분석이 끝나기도 전에 포식자에게 잡아먹혔을 것이기 때문에 우리는 지금 성급하게 결론을 내리고 있다

우리는 우리 앞에 놓인 문제의 모델을 만들고, '현재 우리가 알고 있는 모든 것'이이 모델에 맞아 떨어지는지 확인한다. 그런 다음 이 모델이 틀렸다는 사실을 증명 할 수 있는 방법을 생각해 노력한다.

7.5 우리의 주장에 맞는 현실을 발명하자

- 기계적인 공감이라고 부르는 소프트웨어 설계 접근 방식

하드웨어의 동작 방식을 상당히 깊이 이해해 이를 활용할 수 있도록 코드를 설계했다.

실험을 통해 얻은 중요한 교훈 중 하나는 어리석은 실수를 제거하고 나면 최신 컴퓨터에서 코드의 기본적인 성능에 가장 큰 영향을 미치는 요인은 캐시 적중 실패라는 것이다

캐시 적중 실패를 방지하는 것이 코드의 심각한 고성능 부분에 대한 설계 접근 방식을 지배 하게 되었다

7.6 추측보다는 실험 현실에 입각한 경험주의

- 문제를 해결하기 위해 추측을 내놓고 그 추측이 옳은지 그른지 확인하지 않고 서둘러 구현에 달려들었다.

연구에서 가정한 해결책이 타당하지 않다는 사실을 알 수 있었다. 추측, 가정, 자만심을 바탕으로 진행하는 대신 회의적인 태도로 자신의 아이디어를 확인하는 것은 힘들지만 진정한 진전을 이룰 수 있는 유일한 방법이다. 가장 좋은 시작 방법은 자신이 알고 있다고 생각하는 것이 틀렸을 가능성이 있다고 가정하고 나서, '어떻게 틀렸는지'를 알아내는 방법을 찾아내는 것이다

학계에서는 '병렬성이 높을수록 좋다'고 가정했지만, 이는 동시성 비용이 낮은 가상의 이론적 컴퓨터에 기반한 생각일 뿐이며 이런 컴퓨터는 존재하지 않는다.

이 학자들은 실험주의자였지만 경험주의자는 아니었다. 이런 경험주의의 부족은 학자들의 실험이 잘못되었기 때문에 구축된 모델이 현실의 경험과 일치하지 않았다는 사실을 의미했다.

정리:

순수 과학이 아닌 공학은 해법의 실용성을 고려하게끔 요구한다. 여기서 경험주의가 작동하게 된다

우리는 우리의 추측과 이를 테스트하기 위해 만든 실험에 대해 끊임없이 회의적인 태도를 취하고 현실의 경험과 비교하면서 확인해야만 한다