마스터 링 프로그래밍

En Español.

수년간 마스터 프로그래머를 지켜 보면서 나는 그들의
워크 플로우. 숙련 된 숙련 된 프로그래머를 코칭해온 수년 동안 저는
그 패턴의. 나는 패턴을 도입하는 것이 어떤 차이를 만들 수 있는지 보았다.

다음은 효과적인 프로그래머가 소중한 3e9 초를 최대한 활용하는 방법입니다.
행성.

여기의 주제는 두뇌를 확장하는 것입니다. 숙련공은 해결함으로써 더 큰 문제를 해결하는 방법을 배웁니다.
한 번에 더 많은 문제. 마스터는 해결함으로써 그보다 더 큰 문제를 해결하는 법을 배웁니다.
한 번에 더 적은 문제. 지혜의 일부는 세분화되어 별도의
해결책은 함께 해결하는 것보다 작은 문제가 될 것입니다.

시각

슬라이싱 . 큰 프로젝트를 가져다가 얇게 자르고
문맥. 저는 항상 프로젝트를 더 세밀하게 분할 할 수 있으며 항상 새로운 순열을 찾을 수 있습니다.
다양한 요구 사항을 충족하는 슬라이스.

한 번에 하나씩 . 우리는 효율성에 너무 집중하여
오버 헤드를 줄이기위한 피드백주기. 이것은 어려운 디버깅으로 이어집니다
예상 비용이 우리가 피한 사이클 오버 헤드보다 큰 상황.

실행하고, 올바르게 만들고, 빠르게 만드세요 . (한 번에 하나씩, 슬라이싱 및
쉬운 변경)

쉬운 변경 . 어려운 변화에 직면했을 때 먼저 쉽게 변경하십시오 (경고,
하드) 그런 다음 쉽게 변경하십시오. (예 : 슬라이스, 한 번에 하나씩, 집중,
격리). 슬라이싱의 예.

농도 . 여러 요소를 변경해야하는 경우 먼저 코드를 재정렬하여
변화는 하나의 요소에서만 일어나면됩니다.

결리 요소의 일부만 변경해야하는 경우 해당 부분을 추출하여
전체 하위 요소가 변경됩니다.

기준 측정 . 세계의 현재 상태를 측정하여 프로젝트를 시작하십시오. 이
우리의 엔지니어링 본능에 어긋나는 일을 시작하지만
기준선은 실제로 문제를 해결하고 있는지 알 수 있습니다.

배우기

당신의 총을 부르십시오 . 코드를 실행하기 전에 어떤 일이 발생할지 정확하게 예측하세요.

구체적인 가설 . 프로그램이 오작동하는 경우 정확히 무엇을
변경하기 전에 잘못 생각하십시오. 두 개 이상의 가설이있는 경우
감별 진단.

불필요한 세부 사항을 제거합니다 . 버그를보고 할 때 가장 짧은 재현 단계를 찾으십시오. 언제
버그를 분리하고 가장 짧은 테스트 케이스를 찾습니다. 새 API를 사용할 때 가장 많이 시작하십시오.
기본 예. "그 모든 것이 중요 할 수는 없습니다."
틀렸어.

예 : 모바일에서 버그를보고 컬로 재현

다중 저울 . 스케일 사이를 자유롭게 이동하십시오. 아마도 이것은 디자인 문제가 아니라
테스트 문제. 기술 문제가 아닌 사람 문제 일 수도 있습니다.
이것은 항상 사실입니다].

트랜센드 로직

대칭 . 거의 동일한 것을 동일한 부분으로 나눌 수 있습니다.
그리고 분명히 다른 부분.

미학 . 아름다움은 올라갈 수있는 강력한 그라데이션입니다. 그것은 또한 flout를 해방시키는 그라디언트입니다.
(예 : 많은 기능을 하나의 거대한 엉망으로 인라인).

리듬 . 적절한 순간이 될 때까지 기다리면 에너지가 보존되고 혼란을 피할 수 있습니다. 함께 행동
행동 할 때의 강도.

트레이드 오프 . 모든 결정에는 절충안이 적용됩니다. 무엇을 아는 것이 더 중요합니다
결정은 오늘 어떤 대답을 선택해야하는지 (또는 어떤 대답을
어제 골랐습니다).

위험

재미있는 목록 . 접선 아이디어가 나오면이를 기록하고 신속하게 작업에 복귀하십시오. 재 방문
중지 지점에 도달했을 때이 목록.

피드 아이디어 . 아이디어는 겁에 질린 작은 새와 같습니다. 당신이 그들을 겁 주면 그들은 멈출 것입니다
오고 있습니다. 아이디어가 떠오르면 조금만 먹이세요. 최대한 빨리 무효화
자존감 부족이 아닌 데이터에서

할 수 있지만 자존감 부족의 데이터가 아닙니다.

80/15/5 . 위험이 적고 합리적인 보상 작업에 시간의 80 %를 투자하십시오. 15 % 지출 관련 고위험 / 고수익 작업에 대한 시간. 시간의 5 %를 보상에 관계없이 간지럽 힙니다. 다음 세대에게 80 % 일을하도록 가르치십시오. 에 의해 누군가가 당신의 15 % 실험 중 하나 (또는 덜 자주, 5 % 실험 중 하나)가 성과를 거두어 새로운 80 %가 될 것입니다. 반복.

결론

이 개요의 흐름은 시간을 관리하고 학습을 늘림으로써 위험을 줄이는 것 같습니다.
두뇌 전체를 사용하고 아이디어를 신속하게 분류하여 신중하게 위험을 감수합니다.