

폭포수(Waterfall) 개발팀에서 애자일(Agile) 개발팀으로의 성공적 이행을 위한 5가지 방안

2012. 9. 24. (제26호)

목 차

- I. 들어가며
- II. 애자일 개발팀으로의 성공적 이행을 위한
5가지 방안
- III. 결론

Key Message

애자일 방법론이 많이 도입이 되고 있지만 진정한 애자일 개발팀이 되기에는 많은 과정이 필요함. 애자일 개발팀이 스스로의 문제점을 개선해나가는 자가 조직 팀(Self-organizing Team)이 되도록 애자일에 대해 연구하고, 훈련시키고, 적합한 코치를 고용하는 등 오랜 시간에 걸쳐 많은 노력이 필요함.

■ 들어가며

- 현재 애자일 방법론이 전 세계적으로 많이 도입되고 있는 추세이나, 아직도 많은 회사들이 전통적인 개발방법에 머물러 있음
- 애자일 팀이 비즈니스 요구 사항을 지속적으로 반영하기 위해서는, 적어도 한 달에 한번 이상은 일정한 장소에서 면대면 작업을 수행해야 함
- 애자일 개발팀은 회사에 많은 도움이 되지만 기존의 팀이 애자일 팀으로 변화하는데는 시간, 학습, 원칙 등 많은 과제가 있음

■ 애자일 개발팀으로 변화 또는 시작

- 애자일 팀으로 변화하는데 도움이 되는 몇 가지 프레임워크가 있음
 - Scrum은 수행하기 용이함: 주기(Iteration)를 원하는 기간으로 정하기(일주일단위, 이주일단위 등 용이한 시간으로 설정), 스프린트 계획 회의(Sprint Planning Meeting)를 열기, 스토리 작성하기, 번다운 차트를 작성하고, 언제부터 일별 개발일정을 수행할 것인지 선택하도록 함
- 중요하지만 어려운 점은 팀을 자가 조직 팀(Self-Organizing Team)으로 만들어 고객이나 조직차원에서 팀이 만족할 만한 수준에 도달하는 데는 시간이 많이 걸린다는 것임

1. 전념하기(Under-commit)

- 새로운 애자일 팀은 관리자에게 애자일방식의 장점을 보여주려는데 급급하여 스프린트 후에 전달해줄 결과물에 대해 지나치게 낙관적인 경향이 있음

- 비현실적인 납기일이나 관리자들의 모든 요구사항을 받아들이지 않아야 함
- 결과물에 대한 현실적이고 이성적인 판단이 필요함
- 수준 높은 결과물을 위해 자가 조직 팀(Self-organizing team)을 구성하는 방법이나 많은 새로운 사례들을 연구해야함
- 의도적으로 본연의 낙관주의 버리기
 - 주기에 완성할 수 있다고 생각하는 것보다 적게 작업량을 계획하기
 - 각 스토리에 모든 테스트(자동회귀테스트, 탐색적 테스트 등)를 할 수 있을 충분한 시간을 계산하기
- 며칠간 바쁠 정도로 일을 계획하고 추가적인 스토리를 계획하되 모든 것을 완성하려 하지말기
 - 후에 스크럼의 '전념(Commitment)' 모드를 적용하면 속도가 월등히 빨라짐

2. 애자일에 대해 학습하기

- 기술적 빚(technical debt)¹⁾을 관리할 수 있고 오랜 기간에 걸쳐 성공할 수 있는 핵심 관습이 필요
 - 빠른 피드백
 - 지속 통합(CI: Continuous Integration) 프로세스: 코드를 컴파일하고 배치 가능한 소프트웨어를 만들고 자동화 회귀테스트를 하는 것
 - 자동화 회귀 테스트: 테스트기반의 개발, 자동화 유닛, 통합테스트
 - 자가 조직 팀(Self-Organizing Team): 자체적으로 최근의 사건들을 돌이켜보고 극복할 장애를 파악하여 개선하는 팀으로 스스로 자신의 문제를 파악, 해결하는 것은 어렵기 때문에 이러한 조직을 만들기까지는 많은 시간이 필요함

3. 애자일 팀을 훈련시키고 코치하기

- 팀을 제대로 훈련시킬 수 있는 적합한 인물 찾기
 - 성공적인 애자일 팀을 경험했고 나머지 팀원을 멘토할 수 있는 인물 고용하는 것이 이상적임

1) 기존의 결함들로 인해 새로운 기능을 개발하는데 어려움이 생기는 현상

- 오랜 기간 동안 자주 만나 팀이 제대로 구성되는지 도움받기
- 배움에 많은 시간 할애하기
 - 모든 팀원이 새로운 툴, 새로운 기술, 해당 도메인에 대한 연구 등에 많은 시간 투자해야 함
 - 언뜻 보면 비싸게 보이지만 비즈니스 니즈에 더 빨리 반응할 수 있고 업무효율이 지속적으로 향상되어서 결국 Concept-to-cash 순환주기가 짧아짐

4. 실험을 두려워하지 않기

- 주기마다 적어도 한번은 일을 멈추고 살펴보고 이슈들을 해결하기 위한 실험을 지속적으로 실행하기
 - 큰 장애요인, 팀의 역할 적합성 등을 고려하여 큰 이슈들 파악하기
 - 실험이 완료되면 스토리나 태스크 카드에 적어 다음 주기에 결과 평가하기
- 실험을 작게 유지하여 시간을 효율적으로 활용하여 문제점들을 조금씩 계속 해결해가기

5. 섬기는 리더(Servant leader)들을 발굴하기

- 팀이 자가 조직 팀이 되는 방법을 배우고 스스로 문제를 푼다고 하더라도 장애를 제거하는 것을 도와줄 팀과 직접적인 연관이 없는 제 3의 인물이 필요
 - 스크럼마스터는 코치와 매니저로서 섬기는 리더의 역할을 할 수 있음
- 이러한 리더들은 토론이 잘 진행되게 하고 분쟁을 해소하고, 필요할 경우 고객입장에서 생각하거나 도움을 주는 등 스크럼 프레임워크의 중간에서 일이 잘 진행되도록 도움이 되기 때문에 지속적으로 발굴하는 것이 중요

■ 결론

- 프로젝트는 방법론이나 툴 때문에 성공하거나 실패하지 않고 능력을 최대한 발휘하는 적합한 사람이 있을 때 성공함
- 팀이 점차 발전하도록 새롭거나 나은 방법을 배우는 것을 묵묵히 지켜봐야함
- 매 주기마다 버거운 기술적 빚 없이 수준이 향상된 결과물을 내놓는 과정 자체를 즐기는 것이 좋음

<참고 자료>

1. Planning and Estimating work for an Agile team, GreenHopper 6.0.x,
2. Building Agile Teams, Not SEO Tools., SEER Interactive, 2012