

# 애자일 방법론 vs. 폭포수(Waterfall) 방법론, 10가지 주요 차이점

2012. 7. 30. (제18호)

## 목 차

- I. 폭포수방법론과 애자일방법론
- II. 두 방법론의 10가지 차이점
- III. 결론

## Key Message

폭포수(Waterfall) 방법론은 요구사항을 완벽하게 취합하여 계획을 잘 세우고 그 계획대로 진행하는 방법론이며 애자일(Agile)은 요구사항을 초기에 완벽하게 취합하는 것이 불가능하기 때문에 개발 주기를 반복하고 고객과 소통하면서 소프트웨어의 품질을 발전시키는 방법론으로 이 두 가지 방법론의 차이점을 바탕으로 개발 조직에게 적합한 방법론을 적용하는 것이 중요함

## ■ 폭포수 방법론은 1970년에 창안된 첫 번째 소프트웨어 개발 방법론이며 애자일 방법론은 폭포수 방법론의 과도한 문서업무 때문에 내재하는 낭비를 줄이고자 1990년대에 고안된 방법론임

- 폭포수 방법론은 소프트웨어 개발방법론의 첫 번째 모델로 환영 받았지만 느린 단계별 접근법으로 중복프로세스가 발생하고 릴리즈 간 긴 시간차가 발생함
  - 요구사항 - 분석 - 디자인 - 코딩 - 테스트 - 릴리즈 의 단계를 지님
- 애자일 방법론은 빠르고 모듈식의 주기적인 방식을 통해 개발 전 단계에 걸쳐 요구사항을 지속적으로 분석, 반영하여 릴리즈 간 시간차를 최소화함

## ■ 폭포수와 애자일 방법론의 개발전략의 10가지 기본적인 차이점을 통해 프로젝트에 최적화된 방법론을 적용할 수 있도록 살펴보도록 함

1. 폭포수 : 미리 정의된 요구사항 vs. 애자일 : 프로젝트 과정에 걸쳐 진화하는 요구사항
  - 폭포수는 미리 모든 요구사항을 수집하고 전체적으로 분석과 디자인을 한 뒤 한번에 완성함
  - 애자일은 미리 모든 것을 결코 알 수가 없다는 것을 전제하여 요구사항의 시간적 변화를 인지하고 프로젝트 진행상 지속적으로 요구사항을 반영
2. 폭포수 : 빅뱅(Big Band) 릴리즈 vs. 애자일 : 빠른 릴리즈
  - 폭포수는 요구사항을 제대로 따르면 모든 것이 기술적이고 기능적으로 잘 작동할 것으로 인식하기 때문에 통합을 빨리하거나 자주하지 않음

- 애자일은 작은 크기로 릴리즈 된 기능들을 목표로 하고 있음. 고객이 진정으로 원하는 것을 파악하여 가능한 빨리 고객에게 제품 건본을 제공하는 것뿐만 아니라 고객의 니즈를 제대로 충족시켜주는 것을 중요시 함

### 3. 폭포수 : 계획 중심 vs. 애자일 : 학습 중심

- 폭포수는 일에 필요한 모든 사항들을 완벽하게 측정하여 계획을 수립하며, 각 단계를 단순히 수행하고, 측정에 대한 실제 값만을 기록함
- 애자일은 요구사항을 모두 미리 알 수 없기 때문에 무엇을 만들고 어떻게 할지에 대한 최소한의 지식에서 시작하여 단계를 거듭하면서 최상의 어프로치를 학습함

### 4. 폭포수 : 고객과의 드문 의사소통 vs. 애자일 : 고객과의 지속적인 의사소통

- 폭포수는 기본적으로 문서중심의 접근법으로 이해관계자들 사이에 계약서로 문서를 제공하기 때문에 상호 작용은 최소화하거나 아예 없을 수도 있음
- 애자일은 작업본을 주기적으로 발전시키고 고객과 지속적으로 공유함. 고객과 개발팀은 공동으로 일을 진행해가면서 무엇을 만들지를 함께 구상함

### 5. 폭포수 : 단계별 중간물 전달 vs. 애자일 : 진행하고 있는 작업본을 지속적으로 전달

- 폭포수는 계획중심이기 때문에 일을 진행하면서 진행과정을 보여줄 필요가 있음에 따라, 자세한 상세서의 완성, 컴포넌트 구성 완성, 시스템 통합 완성 등의 단계별 중요한 진척사항들을 프로젝트 단계별로 점검함
- 애자일은 시간이 많이 흘렀지만 특정 기능이 완성되지 않은 소프트웨어 일지라도 프로젝트 초기부터 전체적으로 통합되기까지 지속적으로 진행사항을 공유함

### 6. 폭포수 : 수평적인 단계별로 개발 vs. 애자일 : 기능별 수직 개발

- 폭포수는 소프트웨어 개발을 고층빌딩에 짓는 방식과 같이 생각하여 기초에서 시작하여 단계를 거쳐 튼튼한 기반에 애플리케이션을 만들
- 애자일은 애플리케이션에 어떤 것이 추가되어야 하는지에 대한 요구는 언제나 존재하기에 소프트웨어를 가능한 빨리 개발하고 고객의 피드백을 받아 그것을 반영하는 개발과 피드백 과정을 반복함

### 7. 폭포수 : 프로그래밍은 단순히 공사와 같음 vs. 애자일 : 프로그래밍은 디자인의 확장임

- 폭포수는 애플리케이션 개발에서 프로그래밍을 단순한 공사로 여겨 낮은 입찰가를 제시한 업체에게 의뢰함
- 애자일은 소프트웨어는 개발주기를 거듭할수록 채워지는 틈새가 있기 때문에 디자인과 프로그래밍을 분리할 수 없음

8. 폭포수 : 마지막에 통합 vs. 애자일 : 초기와 이후 잦은 통합

- 폭포수는 최종 결과물이 나오기 전에 통합을 하면 시간을 낭비한다고 여김에 따라 계획대로 정확히 따른다면 모든 것이 완벽하게 작동한다고 가정하고 프로젝트를 수행
- 애자일은 통합이슈와 같은 문제들은 빨리 인지할수록 더 쉽고 싸게 문제점을 해결하기 때문에 초기와 이후에 자주 통합함으로써 문제점을 찾고 바로 고침

9. 폭포수 : 마지막 단계 테스트 vs. 애자일 : 초기와 이후 잦은 테스트

- 폭포수는 마지막 전까지 통합하지 않기 때문에 마지막까지 테스트를 할 수 없지만 초기에 공들여 모든 것을 고려하여 계획을 세웠기 때문에 계획대로 진행된다면 버그는 없을 것으로 기대함
- 애자일은 초기에 통합하는 이유는 잠재하는 결함들과 예상치 못한 사용 패턴 등을 찾을 수 있으며 코드가 적은 초기에 이러한 결함들을 수정하는 것이 쉽고 비용 절감을 기대함

10. 폭포수 : 문서화된 진행 사항 진단 vs. 애자일 : 개발하고 있는 소프트웨어로 진행 사항 진단

- 폭포수는 계획이 애플리케이션 개발에 가장 중요한 측면이기 때문에 계획에 따른 진행 사항을 진단
- 애자일은 애플리케이션 개발은 소프트웨어를 만드는 것이지 소프트웨어를 만드는 방법을 문서화하는 것이 아니기 때문에 소프트웨어가 얼마나 회사의 비즈니스에 부합하는 지로 진단함

## ■ 결론

- 폭포수와 애자일은 소프트웨어 개발의 다른 방법론으로, 그 차이점은 바로 인지하여 조직에 적합한 방법론을 적용하는 것이 중요함

### <참고 자료>

1. <http://www.selectbs.com/analysis-and-design/what-is-the-waterfall-model>
2. <http://www.waterfall-model.com/>
3. <http://www.agile-process.org/>
4. <http://www.ambyssoft.com/essays/agileLifecycle.html>