

테크놀로지 리더를 위한

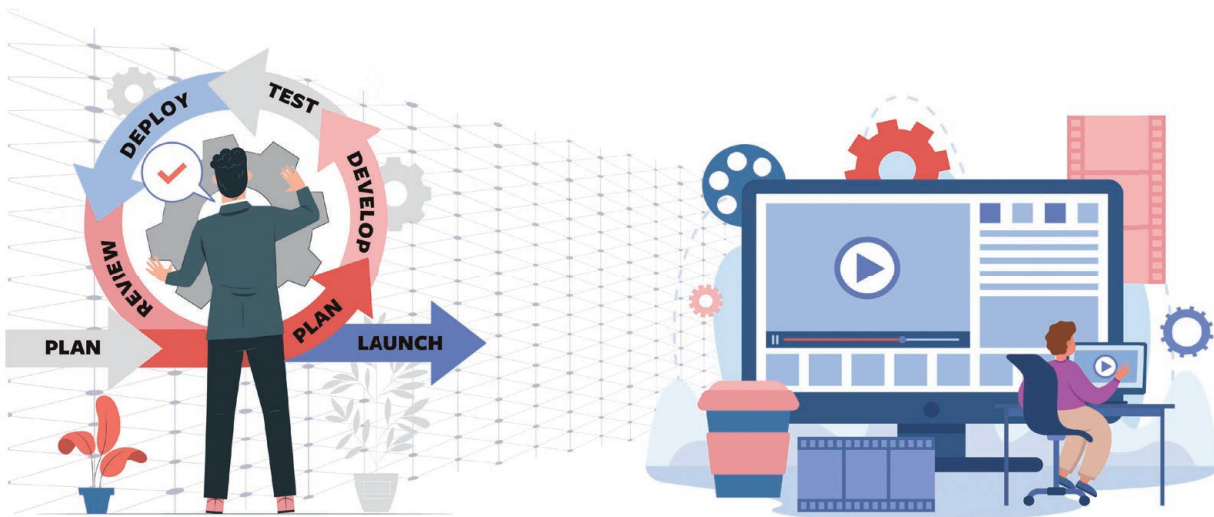
Media & IT(Information Technology)

#5. 소프트웨어 개발과 미디어

강자원

컴퓨터시스템응용기술사

/ KBS MNC(Media Network Center)팀



연재 목차

- | | |
|-------------------------|------------------|
| 1. 소프트웨어공학과 미디어 | 7. 시스템 아키텍처와 미디어 |
| 2. 네트워크와 미디어 | 8. IT 운영과 미디어 |
| 3. 보안과 미디어 | 9. 클라우드와 미디어 |
| 4. 데이터와 미디어 | 10. 인공지능과 미디어 |
| 5. 소프트웨어 개발과 미디어 | 11. 블록체인과 미디어 |
| 6. 애플리케이션과 미디어 | 12. 가상현실과 미디어 |

최근 대학에서는 소프트웨어와 미디어가 융합된 전공이 많아졌다. 한 대학의 전공소개를 보니 “소프트웨어미디어융합전공은 소프트웨어전공과 미디어전공이 융합된 학문으로서 디지털 미디어 및 관련 정치/사회/문화적 현상에 대한 ‘인문사회학적 이해력’, 디지털 미디어를 통해 생산되는 빅데이터에 대한 ‘공학적 분석력’, 미디어/커뮤니케이션 지식과 컴퓨팅 사고력 함양을 토대로 한 ‘창의적 활용력’을 갖춘 인재를 양성한다.”라고 한다.

대학에서 요구하는 인제는 4차 산업시대에 필요한 소프트웨어 리터러시를 지닌 융합형 인재 양성이 키 포인트라 생각된다. 인문사회적인 소양뿐 아니라 IT에 대한 기본적인 이해를 바탕으로 이제는 미디어를 창조적으로 활용하는 인재를 이 시대는 필요로 한다는 의미이다. 그런 의미에서 소프트웨어 개발을 주제로 미디어 트렌드와 함께 이야기해 보고자 한다.

미디어 환경에서의 소프트웨어에 대한 흔한 오해

미디어와 소프트웨어 개발은 어떤 관계가 있을까? 다들 잘 알다시피 미디어 콘텐츠를 제작, 유통, 저장, 보관하는 모든 플랫폼과 시스템에는 해당 목적의 기능을 가진 소프트웨어들이 개발되어 운영되고 있다. 우리는 옷을 살 때, 기성복을 사기도 하지만 내 몸에 맞게 맞추기도 한다. 미디어와 관련된 소프트웨어도 마찬가지다. 예를 들면, 편집용 소프트웨어 같은 경우 Adobe의 Premiere, Avid의 Media Composer 등 정말로 잘 만들어진 프로그램을 도입해 쓰기도 하지만, 유통, 아카이빙, 미디어 자산관리 등과 같이 자사의 고유 시스템에 적합한 프로그램을 위해서는 직접 요구사항을 도출해 개발하기도 한다. 그런데, 방송사 엔지니어들조차 단순히 사용자 입장에서 소프트웨어 개발을 단순한 프로그래밍으로 착각하는 사람들이 많다.

단순 프로그래밍과 창의적인 아키텍처(Architecture)를 만드는 소프트웨어 공학은 구별해야 한다. 고급 빌딩을 건설하는 현장을 생각해 보자. 건설 노동자가 작업을 마무리하기 위해 자신의 노하우를 적용하는 업무와 그 빌딩의 전체 구조를 치밀하게 설계하는 업무는 엄연히 다르다. 건축 설계는 주어진 디자인과 지반 여건을 토대로 안전하고 효율적인 구조를 만들어 내는 전형적인 지식 노동이다. 동일한 건설 업종에 종사하지만, 초점은 완전히 다르지 않은가? 똑같이 컴퓨터를 두드리고 있다고 해서, 같은 언어를 사용한다고 해서, 한통속으로 소프트웨어 개발이라고 간주해서는 안 된다. 이런 착각이 “소프트웨어 개발이라면 나도 해 봤는데...” 하는 인식을 하게 되는 요인 중 하나다. 그러다 보니 소프트웨어 개발을 만만하게 보는 분위기가 있다. “그까짓 것 왜 제대로 못 만들어?”, “그리고, 만드는 데 뭐 그리 비싸?”, “이거 살짝 수정해 주는 게 뭐가 그렇게 어려워?”와 같이 고급 소프트웨어 개발을 해도 존경받지 못하는 이유다.

소프트웨어 개발 방법론

소프트웨어 개발에도 방법론이 있다. 그 방법론 또한 여러 가지인데, 그중 현장에서 가장 많이 사용되는 개발방법론 두 가지에 대해 비교하려 한다.

애자일은 뭐고? 폭포수는 뭐야? Waterfall vs Agile, 이 두 가지는 반드시 알아야 한다.

이 두 가지는 개발 프로세스에서 가장 일반적으로 사용되는 방법론이다. 두 방법론의 주요 차이점은 프로젝트를 구성하고 실행하는 방법이다. 단순히 말하자면, 폭포수는 보다 전통적인 방법론이며 대기업의 프로젝트에서 자주 활용되고, 애자일은 스타트업에 자주 활용되며 소규모 팀에서 애용하는 방법론이다.



	폭포수 방법론	애자일 방법론
요구사항	• 초기 단계에서 명확하게 파악	• 지속해서 요구사항을 반영
특징	• 요구사항 분석 및 문서화에 시간이 오래 소요될 수 있음 • 고객의 요구사항이나 피드백에 대한 대응이 느림	• 문제 해결을 위해 팀원들 간 소통과 협업 중요
적합한 경우	• 요구사항이 매우 디테일하고 명확한 프로젝트 • 변화가 적고 유동성이 낮은 프로젝트 • 큰 규모의 제품 런칭, 메인 기능 개발 등에 적합	• 사용자의 피드백을 수집하고, 빠르게 개선이 필요한 경우 • 요구사항이 변경되거나 문제 발생이 잦은 프로젝트 • 고객, 시장의 변화에 빠르게 대응해야 하는 기업

표 1. 폭포수 방법론과 애자일 방법론의 비교

이 둘은 각각 장단점이 있기 때문에 프로젝트의 성격이나 상황에 따라 선택되어야 한다. 따라서 개발자나 프로젝트 매니저가 어떤 방법론을 선택할지 결정할 때, 각 방법론의 특징을 잘 파악하고 상황에 맞게 선택하는 것이 중요하다.

☑ 개발 프로세스 관점에서 비교

폭포수 방법론은 개발의 각 단계가 순차적으로 진행되기 때문에, 이전 단계의 결과물이 다음 단계에서의 입력물이 되어 전체적인 프로세스의 흐름을 파악하기 어렵다는 단점이 있다. 반면, 애자일 방법론은 개발의 각 단계가 동시에 진행되기 때문에, 개발자들은 개발하면서 프로세스의 흐름을 파악할 수 있다.

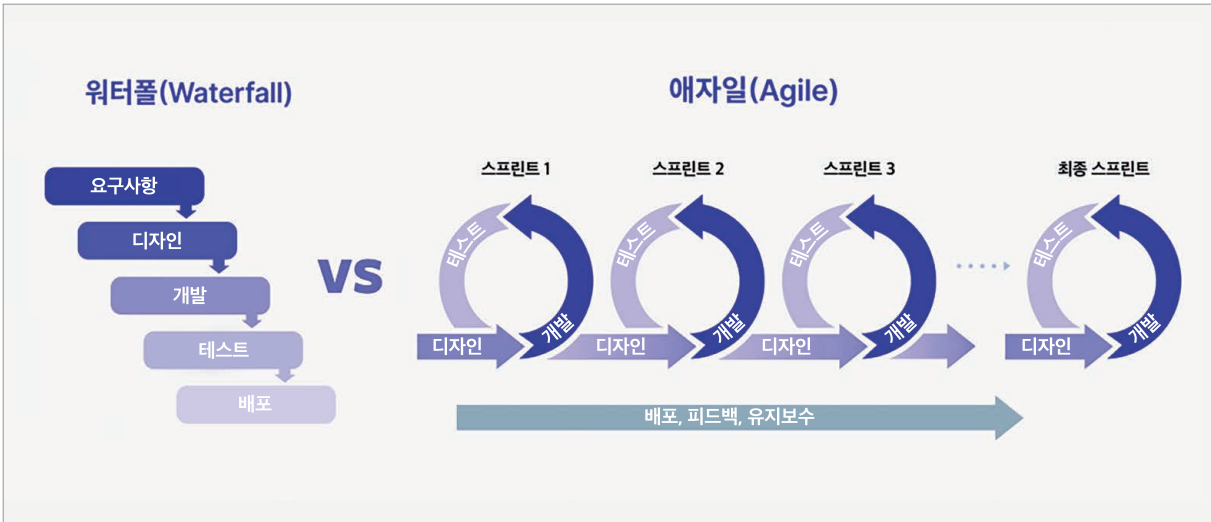


그림 1. 폭포수 방법론과 애자일 방법론의 프로세스

폭포수 방법론은 미리 정해진 요구사항을 바탕으로 기획, 디자인, 개발의 프로세스가 순서대로 진행되어 출시되는 방식의 개발 프로세스로 이전 단계가 완전히 끝나야 다음 단계로 이어진다. 그러니까, 요약하면 **잘 계획하고, 잘 분석하고, 잘 구현하고, 잘 테스트해야 한다**. 이러한 방식은 정해진 시간 내 완벽하게 프로젝트를 구현할 수 있다는 장점이 있지만 각 단계에서 아래로 한번 떨어지면 다시 위로 올라올 수 어렵다는 단점도 가지고 있다.

반면, 애자일 방법론은 민첩성과 행동력이 장점이다. 프로젝트를 진행할 때 우선 진행하고 평가 받고 개선한다.라고 볼 수 있다. 앞을 예측하며 개발하기보다는 일정한 주기를 가지고 변화에 유연하게 대응하고 조금씩 수정해 가며, 실행을 반복해가면서 결과를 조금씩 다듬어 가는 과정으로 개발하는 프로세스를 말한다.

애자일은 옳고, 폭포수는 틀리다?

그럼, 애자일 방법론은 옳고, 폭포수 방법론은 틀린 걸까? 필자는 대학생들과 함께 프로젝트를 매년 다수 진행하고 있는데, 이 친구들이 IT 관련 기업에 면접을 가면 자주 받는 질문이 있다.

〰〰 우리는 애자일 방식으로 일해요 애자일 방식으로 프로젝트를 수행해 본 경험이 있나요? 〰〰

애자일 방법론, 도대체 왜 그렇게 신봉하고 심지어는 ‘조선식 애자일’, ‘K-애자일’이라는 말까지 등장하는 걸까? 그럼, 미디어 업계에서는 주로 어떤 방법론으로 프로젝트를 진행하고 있는 걸까? 대략 1~4주간의 짧은 사이클로 고객의 요구사항을 최대한 반영한 MVP(Minimum Viable Product)를 만들기 때문에 폭포수 방법론보다는 고객의 참여 범위가 비교적 넓은 편에 속하며 고객 가치를 효과적으로 전달할 수 있는 큰 장점이 있다. 시장의 변화에 대응하기 쉽다는 점뿐만 아니라 팀워크 면에서도 개발 과정 속 각기 다른 직무를 수행하는 팀원들의 협력으로 예상치 못했던 시너지가 발생할 수 있다. 이러한 능동성, 유연성 차이로 인해 폭포수 방법론이 엄격하고 규칙적인 이미지의 고리타분한 방법론으로 통하는 것인지도 모른다.

최종 서비스 배포라는 전체적인 그림으로 봤을 때, 기획부터 프로젝트 종료까지 각 단계의 예산과 납기가 명확하게 구분된 폭포수 방법론에 비해 애자일 방법론은 점진적으로 개발해 나가는 방식이기에 최종적인 형태의 전달은 오히려 늦어질 수 있다. 또한, 짧은 개발 주기에서 항상 좋은 품질과 팀워크를 유지하기가 어렵다는 점도 애자일 방법론이 당면한 해결해야 할 과제로 보인다.

미디어 콘텐츠 제작 환경만큼이나 사용자 요구사항이 까다롭고 다양한 프로젝트도 없을 것이다. 일단, 적용해야 할 미디어의 코덱부터 종류가 많고, 각 형식에 따라 호환성도 확인해야 하고... 그러다 보니, 폭포수 방법론을 적용하기엔 예측할 수 없는 이슈들이 많아 모든 것을 단계별로 완벽하게 미리 정의할 수가 없다. 그렇다고 애자일 방법론을 적용하자니 운영을 위해서는 설계도와 같은 각종 산출물이 필수로 있어야 하는데 산출물 작성을 최소화하는 애자일 방법론은 그런 점에선 또 단점이 될 수 있다. 그렇다면, 미디어 환경에서 진행되는 SW 개발 관련 프로젝트들은 어떤 방법론을 적용하는 것이 적합한가? 결론부터 말하자면, 하이브리드 방식을 적용하고 있다.

참 고 용어 설명

● MVP(Minimum Viable Product)란?

최소한의 핵심 기능을 가진 제품을 뜻한다. 최소한의 노력과 개발 공수로 완성할 수 있는 제품이다. 고객의 피드백을 얻어 시장 반응이 어떨지 조사하고 이를 실제 서비스 개발에 적용하는 제품이다.

하이브리드 (Agile/Waterfall Hybrid) 방법론

그러면, ‘하이브리드 방법론’이란 어떤 것을 말하는 것일까? Development Process를 따라서 분석, 설계, 개발, 검증, 인도의 5단계로 구분된다. 단, 개발 단계에서는 애자일 방법론의 스크럼(Scrum) 방식으로 진행하는 Agile/Waterfall Hybrid 방법론(Water-Scrum-Fall)을 말한다.

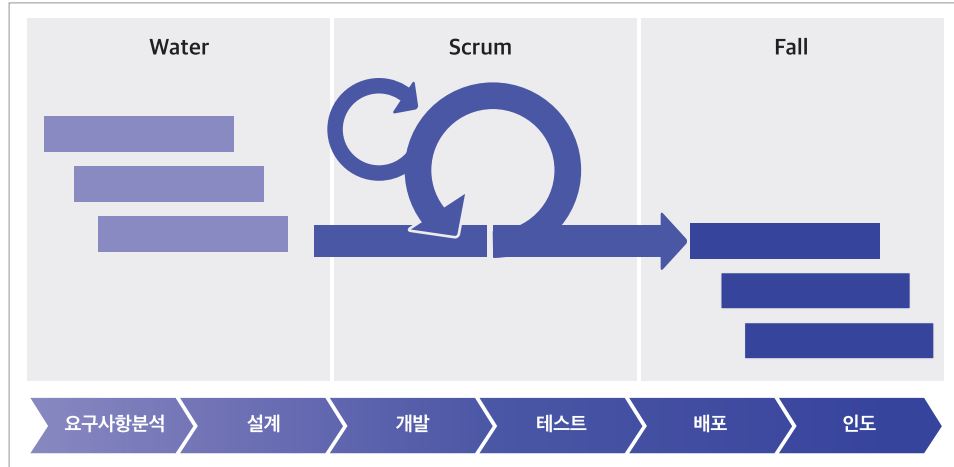


그림 2. 하이브리드 방법론(Water-Scrum-Fall) 프로세스

하이브리드 방법론은 기존 폭포수 방법론에 애자일 방법론을 일부 결합하여 프로젝트 상황에 맞게 활용하는 것이다. 문서작업을 어느 정도는 유지하되 프로젝트 변화에 대응하기 위해 협업과 작업을 반복하는 방식이 대표적이다. 기본적으로 폭포수 방법론의 각 단계의 연결점에서 서로 협업하면서 단계별로 반복을 허용하는 방식이다. 이렇게 프로젝트를 진행하게 되면 이전 단계에서 제기되었던 변경된 요구사항을 다음 단계에 적용할 수 있으며 최종 릴리즈하기 전 베타테스트를 거치며 짧지만 애자일의 스프린트를 가질 수 있다. 마지막엔 베타 오픈을 통해 수집된 피드백을 유연하게 대응하고 반영하여 최종 릴리즈하게 된다.

Hybrid 방법론은 [그림 3]과 같이 크게 2개의 phase로 나누어진다. 마지막 단계인 Release(Open) 단계를 폭포수 또는 애자일 방법론 둘 중 어떤 것을 선택하는지는 프로젝트의 특성 또는 조직문화에 따라 선택적으로 변경 적용하면 된다.

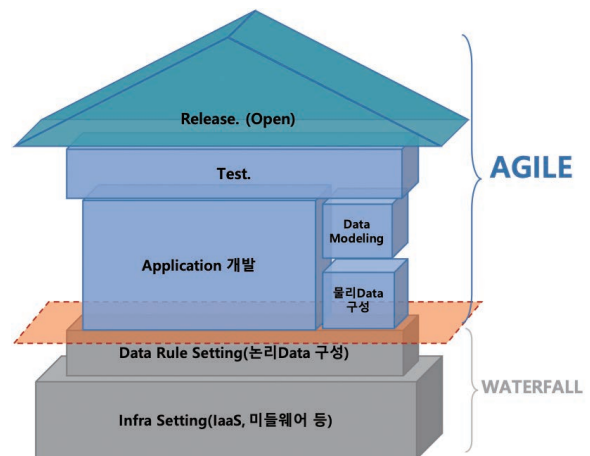


그림 3. 하이브리드 방법론 아키텍처

참 고 용어 설명

● 스크럼(Scrum)이란?

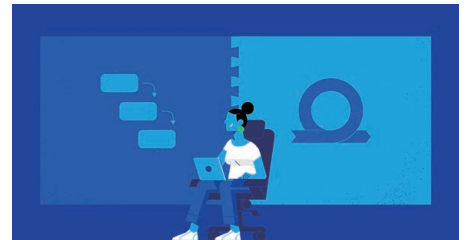
‘스크럼(Scrum)’의 사전적 의미는 ‘럭비에서 반칙이 있을 때, 양편 선수가 밀집하여 서로 팔을 꼭 끼고 뭉치는 일. 그 사이로 굴러 넣은 공을 자기편 쪽으로 빼냄’의 뜻을 가지고 있다. 팀이 중심이 되어, 개발의 효율성을 높이는 것을 목적으로 한 애자일 방법론의 하나이다.

첫 번째 단계는 폭포수 방법론에 기반한 프로젝트 인프라 조성 작업이다. 프로젝트 초기 20~30%의 기간 동안 프로젝트 전반에 영향을 미치는 기반 작업을 우선 수행하는데, 여기에는 Cloud(IaaS 포함), 미들웨어, Application Architecture 작업은 물론 데이터 룰 세팅과 같은 논리 데이터 작업이 포함된다. 이때, 화면설계(UI/UX 표준) 또한 세팅된다. 여기에 프로젝트 관리를 위한 최소한의 산출물까지 작성이 완료되면 더욱 완벽하다.

두 번째 단계로, 애자일 방법론에 기반한 Application 개발이다. 프로젝트 기간 중 70~80%의 기간 동안 실질적인 개발 작업을 수행하는데, 구체적으로는 Application 개발과 함께 데이터 모델링, 물리 데이터 구성 작업을 Sprint별로 진행한다. 각 스프린트 기간에 개발, 테스트, 리뷰, 회고 등 애자일 방법론을 이루는 활동은 모두 포함된다. 스프린트별 개발이 마무리되면 통합테스트 및 안정화 작업 이후에 Release 및 배포는 프로젝트의 규모 조직의 특성 등에 따라 애자일 방법론이 적용되기도 하고 폭포수 방법론이 적용되기도 한다.

그럼, 하이브리드 방법론이 정답인가?

폭포수 방법론과 애자일 방법론의 장점을 차례로 진행한다는 전제하에, 우선 프로젝트의 기초를 단단히 다져놓고, 그 위에서 개발자들이 능력을 발휘할 환경을 조성한다는 점에서 효율적이라 볼 수 있다. 좀 더 명확하게는 개발자들에게 최대한의 시간을 보장해줄



수 있다는 점에서 하이브리드 방법론의 장점을 찾을 수 있다. 또한 첫 번째 단계(폭포수 방법론 적용) 동안 고객이 원하는 산출물 작업을 마무리하거나 최소한 템플릿들을 지정해 놓는다면 개발자들이 부담스러워하는 ‘개발 중 산출물 작성’의 고충을 크게 줄여 줄 수 있다. 개발 프로젝트를 진행하면서 “이러저러한 산출물이 누락됐다.”, “비즈니스 다이어그램 주세요.”, “애플리케이션 아키텍처 다이어그램은요?” 등 요구하지 않으면 챙겨 받지 못하는 산출물들이 많고, RFP에 상세히 명세 해 두지 않으면 요구할 수도 없다. 당연한 것이 아니기 때문이다. 그래서 사업 진행 시 운영조직을 고려해 산출물을 잘 챙겨둬야 유지보수가 원활할 수 있다.

그런데, 이렇게 명시해 두어도 개발업체가 영세해서 또는 개발 납기 준수를 지키려 하다 보니 개발 산출물은 개발자에게 짐이 될 수밖에 없다. 이처럼 개발을 위한 가용한 시간을 늘려주고 특히 집중을 방해하는 요인을 제거한다는 점에서 하이브리드 방법론은 애자일 방법론으로만 진행할 때 발생하는 ‘현장의 아쉬움과 불만’에 대한 대안으로써 충분히 주목받을 만하다.

사실, 미디어 콘텐츠 제작 현장이 IT와 융합하면서 방송 엔지니어들은 이러한 개발방법론을 전혀 모른 채 미디어 관련 솔루션 개발업체들과 협업하고 사업을 진행해 왔을 것이다. 그리고는 수많은 시행착오를 겪고 많은 갈등이 있었을 텐데 그 과정 중에 어쩌면 적당한 거리와 합의를 통한 적절한 방식을 이렇게 자연스럽게 찾은 건 아닐까 생각된다.

필자가 본격적으로 IT 프로젝트를 진행하면서 느낀 점은 엔지니어들은 지금 우리가 사용하고 있는 이 방법론이 ‘하이브리드 방법론’이라는 걸 알고 진행하는 것일까? 하는 의문점이 있었다. 이번 원고를 통해 우리가 현재 현장에서 적용해 사용하고 있는 방식이 무엇인지 알 수 있는 계기가 되었길 바란다. 📖