

<http://dx.doi.org/10.7236/IIBC.2015.15.6.147>

IIBC 2015-6-21

Agile 방법론을 이용한 S/W개발 프로세스 및 성숙도 측정

Measurement of S/W Development Processes and Maturity using Agile Methodologies

김태달*

Tai-Dal Kim*

요 약 최근 애자일 개발프로세스가 생산성 증대효과, 품질향상, 고객만족 등의 효용성이 입증되면서 프로젝트 추진 시 방법론 선택에 있어 전문가들에게 많은 관심을 갖게 되었다. 시대적 요구와 사용자 요구에 부응하기 위한 방법론 선정과 관련하여, 본 논문에서는 제품 기반의 Cross functional team의 문제점을 해결하고자 제안된 방법론이 Feature Team 모델이며, 이 모델이 Cross functional team을 조직하고, 이 팀이 결과물(프로덕트) 단위가 아니라, 기능단위로 여러 프로덕트에 걸쳐서 개발을 진행하는 모델이라는 점에 대해 조사하였으며, 가치-주도형 애자일 기법을 통해 계획-기반형 모델과 차이점을 제시하였다.

Agile 성숙도 측정기준을 검토하였으며, PRINCE2 Agile Health-check 항목 연구를 통해 Agile 기법이 프로젝트 착수단계의 요구사항은 확정 본이 아닌 언제라도 변경될 수 있다는 것을 전제를 해서 객관적인 평가를 받는 방법에 대해 연구하고 프로젝트 팀 구성에 대해 향후 발전방향을 제시하였다.

Abstract Recently agile development process became increasing effectiveness, quality, attention to experts in customer satisfaction, as evidenced in this methodology when selecting projects promoting productive efficiency.

With regard to contemporary needs and user requirements on the methodology selected to meet this paper is the product based Cross functional team suggested methodology Feature Team model to solve problems of this model, and organizing the Cross functional team, this team but this outcome (product) basis, were examined for the model that points to progress the development across multiple product as a functional unit, value-driven agile project through the Skills-based model and proposed a difference. And it examined the Agile Maturity metrics.

PRINCE2 Agile Health-check entries future development direction of Agile techniques is a requirement of the project outset has studied the subject objective evaluation by the assumption that they can be changed at any time, not fixed this way and for the project team through research The proposed

Key Words : PRINCE2, eXtreme Programming(XP), Scrum, Crystal Family, Feature-Driven Development, Adaptive Software Development(ASD) eXtreme modeling, Agile Health-check, Ruby on Rails

1. 서 론

국제적으로 소프트웨어 산업분야도 많은 변화를 가져오고 있다, 90년대 후반까지 사용했던 개발방법론과 관

리방법은 고객의 요구사항의 변경에 따른 작업량 예측이 매우 어려웠다. 고객의 요구는 대부분 매우 유동적이고 다양해지면서 고전적인 소프트웨어공학이나 관리기법만으로는 대처할 수 없게 되었다. 이러한 문제점을 해결하

*정회원, 청운대학교 컴퓨터학과
접수일자: 2015년 10월 12일, 수정완료: 2015년 11월 12일
게재확정일자: 2015년 12월 11일

Received: 12 October, 2015 / Revised: 12 November, 2015 /

Accepted: 11 December, 2015

*Corresponding Author: ktd@chungwoon.ac.kr

Dept. of Computer Engineering, Chungwoon University, Korea

기 위한 해결책으로 납기준수는 물론이고 고객의 요구조건에 적합하고 품질 좋은 결과물 양산과 재사용을 용이하게 하기 위해서 적용되던 방법이 객체지향방법론이다. 그러나 사용자 요구 조건에 부합한 객체지향 개발을 위한 개발프로세스가 필요하게 되었는데 이러한 맥락에서 만들어진 것이 애자일 개발프로세스이다.

본 논문에서는 애자일(Agile) 개발방법론을 사용하려고 하는 조직을 위해 애자일 적용에 있어 현재의 공통적인 문제점, 애자일 장애요인, 애자일의 성공기준, 애자일 개발프로세스와 전통적인 개발 프로세스와의 차이, 애자일 개발방법론에 대해 조사 연구하였다. 그리고 애자일 성숙도 연구에서는 태도측면 점검항목, 환경측면 점검항목, 프로세스측면 점검항목, 기법측면 점검항목에 대해 조사하였고, 마지막으로 영국 정부에 의해 제정된 경험을 바탕으로 구조화된 방법론이며, 프로젝트관리 국제표준으로 사용되고 있는 PRINCE2 (an acronym for PProjects IN a Controlled Environment, version 2)의 점검항목과 세부내용에 대해 연구하였다.

II. 연구 배경

프로젝트 성격이 최근 국내. 외적으로 범위, 일정, 비용, 품질, 인적자원에 대한 기준이 전통적인 방식에서 크게 변화가 있는 것이 사실이다

일정측면에서도 무조건 납기 우선이 아닌 사용하기 적합한 시점에 인도되어야 한다는 점, 범위측면에서도 최초 정의된 사양과 기능에 역매인지 않고 이해당사자의 실제적인 요구사항의 변경이 수시로 적용되어야 한다는 점, 예산측면에서도 예산 내 집행이 아닌 투자수익률(ROI: Return On Investment)이 도출되어야 한다는 점, 품질측면에서도 일정과 예산에 무조건 맞추어 고품질의 결과물을 받기 원한다는 점, 투입요원측면에서도 일정과 예산에 맞추기 위해 육체적 또는 정신적으로 일체감이 이뤄야한다는 점, 환경이 갖추어진 상태에서 일을 해야 한다는 점이다. 이러한 관점에서도 방법론 적용에 있어서 변화를 요구하고 있는데, 그 대안으로 제시되는 것이 애자일(Agile)방법론이다.

애자일방법론이 모든 프로젝트에 적용할 수 있는 방법론이 아님을 전제하고, 프로젝트를 추진할 때 그 활용 범위에 대해 연구를 하게 되었다.

III. 이론적 고찰

애자일 적용의 공통적인 문제점, 애자일 장애요인, 애자일의 성공기준, 애자일 개발 프로세스와 전통적인 개발 프로세스와의 차이, 애자일 개발방법론에 대해 조사 연구한다.

1. Agile 적용의 공통적인 문제점

첫째. 애자일(Agile)을 적용하였지만 어떠한 성과도 내지 못한다고 생각하는데 문제점이 있다.

스크럼 프랙티스(Scrum practice)를 적용하지만 별도 도움이 안 되고 오버헤드만 발생하고, 이 방법론은 관리자를 위한 것이지 팀에게는 별 도움이 안 된다고 생각하는데 문제점이 있다.

둘째. 애자일(Agile)을 적용하고 성과는 있지만 생각했던 만큼 효과가 없다고 생각하는데 문제점이 있다.

기업 이미지 통합전략 CI(corporate identity) 정도는 도움이 되었지만, 그 외에는 그다지 도움이 안 된다고 느끼는 점.

셋째. 팀이 애자일(Agile)을 적용할 때 조직에서는 어떠한 도움도 제공하지 않으며 부정적 시각으로 바라본다는데 문제점이 있다.

조직에서는 이런 문제들을 해결하기 위해 프로젝트 혹은 팀의 특성과 관계없이 일률적인 프로세스와 체크리스트를 가지고 애자일(Agile)을 적용하려함으로써 많은 팀의 반발을 가져오게 된다.

조직에서는 팀에게 내. 외부 전문가를 활용하여 애자일 전문교육을 제공하나 팀이 갖고 있는 고유의 문제를 해결하지 못하는데 문제점이 있고, 심지어 어떤 조직은 애자일이 부적합하다고 생각하고 애자일 적용을 중단하기도 한다. 그리고 한번 실패하면 애자일(Agile)을 다시 시도하려고 하지 않는다는데 큰 문제점이 있다.

2. Agile 장애요인

개발 프로젝트 팀에 만연되어 있는 현재의 사고에 일대 개혁이 있지 않으면 애자일 방법론 적용은 매우 힘이 든다. 다음은 현장에서 일어나고 있는 애자일 장애요인 사례의 좋은 예이다.^{[1][2][3]}

- 1) 현재 사용하는 개발방법론과 프로세스가 완벽하고 체계적이므로 변경할 필요가 없다.

- 2) 애자일을 기술적으로만 접근하려고 하고, 애자일이 가지고 있는 전반적인 사상을 이해하지 못하고 있다.
- 3) 기존 경영자 및 리더가 명령과 통제에 익숙해 있고, 마이크로 관리(Micro management) 중심의 리더십에서 벗어나지 못하고 있다.
- 4) 애자일(Agile)을 제대로 연구하지 않고 더군다나 교육이나 코칭도 받지 않고 나름대로 해석하여 적용하고 있다.
- 5) 분업화된 역할과 개인의 성과중심의 평가문화에 익숙하지 못하는 문제점을 안고 있다.

3. 애자일(Agile) 개발프로세스와 전통적인 개발프로세스와의 차이

폭포수 개발방식과 애자일방법론은 계획된 프로세스와 경험적 프로세스로 구분된다.

계획된 프로세스는 요구사항이 초기에 정확히 도출되고 변경이 없다는 가정 하에서는 상당히 이상적인 개발방식이 될 수 있지만, 소프트웨어의 비가시성(Invisibility)과 변경용이성(Changeability) 및 요구사항의 불확실성(Uncertainty) 등의 특성으로 인해 현실적으로 고객의 눈높이에 맞는 만족스러운 소프트웨어 개발이 어렵고 프로젝트 중간에 결과물을 볼 수 없다는 단점을 내포하고 있다.

애자일 방식의 경험적 프로세스는 고객의 요구에 맞추어 출시(Release)와 반복(Iteration) 개발을 거듭해 가면서 이전 경험을 다음 프로세스에 반복해 가며 바로 적용함으로써 소프트웨어 품질향상과 소프트웨어 인도 시점을 고객의 요구시점에 맞출 수 있다는 장점을 갖고 있다.

표 1. 폭포수 개발방법론과 애자일과의 차이
 Table 1. the difference between agile and waterfall development methodologies

구분	Agile 방법론	기존 방법론
요구사항 관리	지속적인 요구사항 개발 및 변경 수용	초기 요구사항 수집 및 엄격한 변경관리
계획수립	두 단계 계획(짧은 계획 수립 및 경신), 경험 기반 프로세스	상세한 계획 수립(up-front), 계획 기반 프로세스
설계	적시(just-in-time) 설계	상세한 사전(up-front) 설계
문서화	경량(Lightweight) 프로세스 및 문서화보다 코드를 강조	중량(Heavyweight) 프로세스 상세한 문서화 강조
역할	전체 팀(Whole team) 워크를 중요	엄격한 역할 분리

프로젝트 착수 후 일반적으로 이해 당사자 간에 통제 불능상태로 직면하게 되는 경우도 발생하는데, 이런 경우 문제를 해결하기 위해 Agile 이라는 방법론이 그 대안을 제시하고 있다. 고객과 개발자 간에 협력적으로 프로젝트를 추진하도록 협업 형태로 개발모형을 전개하면서 서로가 독려하고, 가능하면 계획범위 내에서 적용하게 하는 기법이 기존방법론의 계획-주도형이 아닌 가치-주도형 기법이라 할 수 있다.

애자일 개발프로세스란 어느 특정 개발방법론을 가리키는 말이 아닌, 애자일(Agile: 기민한, 좋은 것을 빠르고, 낭비 없게 만드는 것) 개발을 가능하게 해주는 다양한 방법론 전체를 적용할 수 있음을 의미한다.

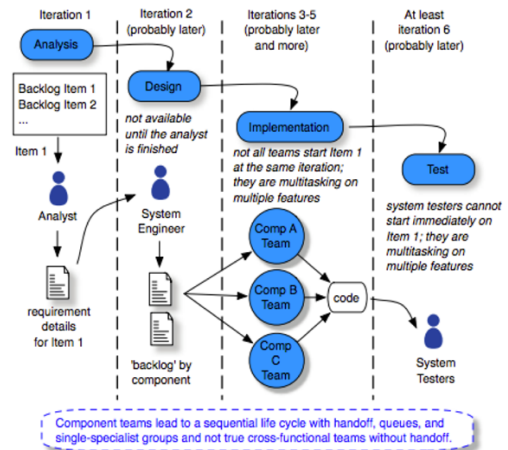


그림 1. Component(Product) 기반 개발팀의 개발 프로세스
 Figure. 1. Component (Product) based on team development processes

전통적인 개발프로세스와의 차이점이라면, 고객과의 협업, 즉 고객과의 피드백을 통한 의사소통으로 고객의 요구조건에 대해, 적시 적소에서 빠르게 대응하여 결과적으로 문서가 아닌 고객이 원하는 결과물로 즉 프로그램을 보여주면서 완성도 높은 결과물/프로그램을 만든다는 데서 큰 차이점이 있다.

그래서 그림 1과 같이 완성도와 개발기간 단축을 위해 재사용을 고려하게 되었고 컴포넌트 기반 개발프로세스에 관심을 갖게 되었다.

요약하면, 프로세스와 개발보다는 사람과의 의사소통에 중점을 두고, 기획하고 문서화보다 눈에 보여 지는 결과물에 중점을 두고, 계약에 억매인 협상 보다는 고객과

의 사전 협업을 통해 해결하고, 계획에 대한 맹종하기 보다는 변화에 대해 즉각 대응한다는 점에서 차이가 있다.

4. 애자일(Agile) 개발 방법론

애자일(Agile) 개발프로세스로 불리는 대표적인 개발 방법론은 다음과 같은 종류가 있다.^{[4][5]}

1) 익스트림 프로그래밍(eXtreme Programming, XP)

켄트 벡(Kent Beck) 등이 제안한 소프트웨어 개발방법이며, 사용자의 요구사항이 시시각각 변동이 심한 경우에 적용하기에 적합한 개발방법이다.

1999년 켄트 벡의 저서인 'Extreme Programming Explained - Embrace Change'에서 발표되었다.

애자일 개발 프로세스의 대표자로 애자일 개발프로세스의 보급에 큰 역할을 하였다. 이 방법은 고객과 함께 2주 정도의 반복개발을 하고, 테스트와 우선 개발을 시도하는 명시적인 기술과 방법을 가지고 있다.

2) 스크럼(Scrum)

일본 히토츠마시 대학의 노나카 이쿠지로와 타케우지 히로타카가 1986년 1~2월 Harvard Business Review에 올린 "The New Product Development Game"^[1]에서 시작된다. 그 후 1991년 디그라스(DeGrace)와 슈탈(Stahl)이, "Wicked Problems, Righteous Solutions"^[2]에서 스크럼을 처음 언급했다. 처음 노나카와 타케우지가 스크럼을 만들 때의 목표는 공업품의 개발이었으나, 1995년 Ken Schwaber가 이 방법을 Advanced Development Method라는 이름으로 자신의 회사에서 사용하였다.

스크럼(Scrum)은 프로젝트관리를 위한 상호, 점진적 개발방법론이며, 애자일 소프트웨어 공학 중의 하나이다. 스크럼(Scrum)은 소프트웨어 개발 프로젝트를 위하여 고안되었지만, 소프트웨어 유지보수 팀이나 일반적인 프로젝트/프로그램 관리에서도 적용될 수 있다. 30일마다 동작 가능한 제품을 제공하는 스프린트(sprint)를 중심으로 하고 있다. 매일 정해진 시간에 정해진 장소에서 짧은 시간의 개발을 하는 팀을 위해 개발된 프로젝트관리 중심의 방법론이다.^{[6][7][8]}

3) 크리스털 패밀리(Crystal Family)

프로젝트의 규모와 영향의 크기에 따라서 여러 종류

의 방법론을 제공한다. 그중에서 가장 소규모 팀에 적용하는 크리스털 클리어는 익스트림 프로그래밍(eXtreme Programming)만큼 엄격하지 않고 효율도 높지 않지만, 프로젝트에 적용하기 쉬운 방법론이다.

4) Feature-Driven Development

feature마다 2주정도의 반복 개발을 실시한다.

Peter Coad가 제창하는 방법론으로써, UML(Unified Modeling Language)을 이용한 설계기법과도 밀접한 관련을 가진다.

5) Adaptive Software Development(ASD)

소프트웨어 개발을 혼란 자체로 규정하고, 혼란을 대안체로 그에 적응할 수 있는 소프트웨어 방법을 제시하기 위해 만들어진 방법론이다. 내용적으로는 다른 방법론들과 유사하지만, 합동 어플리케이션 개발(Joint Application Development, 사용자나 고객이 설계에 참가하는 개발 방법론)을 사용하고 있는 것이 조금 다르다.

6) 익스트림(eXtreme) 모델링

익스트림 모델링은 UML을 이용한 모델링 중심 방법론이다. 다만, 다른 모델링 방법들과의 차이점은, 언제나 실행할 수 있고 검증할 수 있는 모델을 작성하는 공정을 반복해서 최종적으로는 모델로부터 자동적으로 제품을 생성하게 한다.

IV. Agile 성숙도 판별 기준

애자일을 시행하고 있는 기업을 보고 성숙도가 어느 정도 인지를 점검하기 위해서는, Agile 가치 및 원칙을 준수하면서 하는지?, 아니면 Daily Stand-up을 하는지?, Scrum/ Kanban 기법만 사용하는지? 에 대해서는 그림 2와 같이 다음 3가지만 확인하면 된다.^[9]



그림 2. Agile 성숙도 측정 기준
Figure. 2. Agile Maturity metrics

첫째, 에픽(Epic) >사용자 스토리(User Story) >백 로그(Backlog) >완료의 정의(Definition of Done)가 아주 명확하게 정의되어 있어야 한다. 특히, 완료의 정의 즉, 검수조건은 작업/백로그(Task/ Backlog)의 완결성을 확인하는 주요 측정단위이다.

애자일(Agile)을 실행하는 조직에서 Task/Backlog/FBS: 전진기지 시스템(forward based system)만 있고, 완료의 정의(Definition of Done)/검수조건이 없다면 이는 애자일을 흉내만 낸 것이다.

둘째, 이전 단계의 결과와 사용자 요구를 즉시 적소에 반영하는 회고 문화가 주기적으로 수반되어야 한다. 회고/교훈은 프로젝트관리 성숙도를 높일 수 있는 필수 도구이다.

팀원 모두 돌아가면서 일전의 스프린트/반복(Sprint /Iteration)에 대해서 활용점 회고와 개선점 회고를 전체적으로 평등한 의사소통 상태에서 이루어져야, 개인은 물론이고 조직과 프로젝트에서도 긍정적인 영향을 미치고 개선을 할 수 있다.

셋째, 속도(Velocity) 관리이다. 이때 주의해야하는 것은 애자일(Agile) 수행 1년에서 2년 기간은 속도매트릭스(Velocity Matrix)를 구축하는데 역량을 쏟아야지 속도를 평가하면 안 된다.

초기 프로젝트 업무/Backlog 유형 라이브러리 및 개인 분야별 역량 라이브러리를 기준으로 회사/조직에 맞는 성숙도 속도 데이터베이스(velocity Database)를 구축하는데 힘을 쓰고, 이후에는 초기 예측한 속도와 실제 수행된 속도의 격차(gap)의 근본원인과 문제해결에 중점을 두어야 한다. 이것이 평가에 반영되려면 1년차는 DB구축, 2년차는 시험 운영, 3년차는 속도 평가가 의미가 있다.

애자일(Agile)은 회사의 전략적 포트폴리오, 프로그램, 프로젝트관리와 조직 역량관리를 바탕으로 장기적인 안목을 가지고 문화 개혁운동으로 접근하는 것이 올바른 방식이다. 요약하면, 애자일 가치와 원칙을 제대로 이행하려면 점진적으로 조직의 비전을 생각하면서 환경을 만들어 나가는 방향으로 이끌어 가야 한다.

V. PRINCE2 Agile 체크리스트

애자일(Agile) 성숙도 판별기준을 위해 자가 진단 점

검할 수 있는 애자일 건강 체크(Health-check) 방법을 제시하고 있다.

Axelos Prince2 Agile^[10] (Agile Project Management Best Practice)에서 애자일을 올바르게 수행하고 있는지에 대한 항목으로 프로세스와 기법뿐만 아니라, 태도/환경이 포함되어 있는 점이 폭포수 (Waterfall)모델과 가장 차이가 있는 부분이다.

애자일을 수행하고 있는 조직에서는 아래 건강정도 체크를 통해서 자체 진단을 할 수 있다.

1. PRINCE2 애자일 건강정도 체크(Health check)

PRINCE2 (an acronym for PRojects IN a Controlled Environment, version 2)는 영국 정부에 의해 제정된 프로젝트관리 국제표준으로, 통제 가능한 환경에서의 다양한 프로젝트와 수많은 이해관계자(스폰서, 프로젝트관리자, 프로젝트 팀, 학계, 컨설턴트)의 경험을 바탕으로 구조화된 방법론이다.

가. 태도 측면 점검항목

- 팀은 정기적으로 가치와 이익에 대해서 이야기 나누고 있다.
- 사물을 간단하게 보는 태도를 가지고 있다.
- 모든 이해관계자는 프로젝트의 최신 정보를 갱신 받고 있다.
- 이사회는 프로젝트에 관계된 사람들과 주기적으로 상호소통 하고 있다.
- 협력과 신뢰, 비난 없는 문화를 가진다.
- 프로젝트 관리 팀에 의해 자기 주도적 문화가 지원되고 있다. 검사와 적용 아울러 지속적 개선을 위한 문화가 형성되어 있다.
- 투명성이 대세를 이루고 있다.
- 상호 협력에 의해 기획이 점진적으로 구체화 되어가고 있으며 업무가 할당되고 있다. 화기에애한 분위기와 유머가 잘 유지되고 있다.

나. 환경측면 점검항목

- 프로젝트에 투입된 사람들이 행복해하고, 재미있게 일하고 있다.
- 팀이 안정적으로 보호막 안에서 간섭 없이 상호 협력적으로 일을 하고 있다.

- 프로젝트의 변화에 대해 익숙하고 민첩하게 대처한다.
- 고객이 모든 단계에 대해 참여하고 관여한다.
- 프로젝트 현안에 대해 고객관점에서 다양한 시각으로 접근한다. 도출된 결론 혹은 산출물은 모든 사람에게 명쾌하다.
- 피드백에 대해 갈망하고, 고객이 진정으로 무엇을 원하는지를 수집하기 원한다.

다. 프로세스 측면 점검항목

- 애자일(Agile) 학습이 조직 전체에 대해 확산되고 있다. 의사소통이 매우 편잡고 빠르다.
- 사람들에게 권한이 위임되고, 통제가 가볍게 되어가고 있다.
- 의무사항은 철저히 이행하여야 한다.
- 프로젝트 팀과 운영 팀은 애자일을 이해하고 있다. 이것은 애자일을 하는 것과는 다른 의미이다. 프로젝트에 임하는 사람들의 마음가짐이 전반적으로 애자일을 수행하고 있다.
- 주기적 출시(release)는 실제 운용상 사용을 목표로 하고 있다.
- 무엇이 고정적이고 유연한 것인지를 결정하는 5요소(테드라인, 품질수준, 변화수용, 팀 안정성, 고객은 모든 것이 필요치 않음)에 대해 잘 알고 있다.
- 주기적 발행으로 인해 심각한 일들이 일상적으로 발생할 수 있음을 인지하고 있다. (release/sprint 리부).
- 모든 역할은 명쾌하게 정의되었고 이해되었다.
- Waterfall(PRINCE2) 이 애자일로 해석되기 시작한다.
- 예외사항 처리가 철저히 관리된다.
- 통제 수준은 불확실성 수준과 기준으로 맞춘다.
- 작업패키지(Work package) 가 잘 구성되었으며, 명쾌하기 때문에 의사소통에 문제가 없다.
- 요구사항과 사용자 스토리(user story)가 숨겨진 것 없이 잘 기술되어 있다.
- 일하는 방식은 이익이 우선이라는 점에 대해 주기적이고 점진적으로 전달되고 있다.
- 프로젝트 기간 내내 시종일관 동일한 패턴으로 일을 한다.
- 명확하게 완료(done), 준비(ready)를 식별하고, 업무 합의과정이 투명하다.

- 기획과 업무하는 방식이 기능중심과 시간제약 방식이다.
- 최소 실행 가능한 제품(Minimum Viable Product)가 모두에게 투명하다. 그리고 프로젝트 역할에 대한 이해와, 학습에 의한 개선방식을 이해하고 있다. 품질 확인과 시험은 독립적인 방식으로 진행한다.
- 품질 확인방식으로 개선된다. 그리고 이것은 제품의 형상을 구체화하는 방향으로 진행된다.
- 기획은 단기, 중기, 장기 계획을 포함한 모든 기획요소들과 병행하며, 불확실성에 대한 요소를 고려한다.
- 경험에 의해 실증적인 기획이 이뤄진다.
- 초기 요구사항은 불필요하게 상세화 하려는 과정을 피하는 형태로 수행된다. 그리고 프로젝트 보증을 이러한 민첩성에 기반을 두고 가치를 높이는데 중점을 둔다.

라. 기법측면 점검항목

- 위험요소(risk)는 때로는 Spike, 프로토타입과 실험에 의하여 완화하고 있다.
- 차트 작성(Burn charts)는 진척사항을 명확하게 보기 위하여 다양하게 사용된다.
- 사용자 스토리(User story)는 대화와 의사소통을 활발하게 유도하고 있다.
- 칸반(Kanban)은 단지 칸반 보드뿐만 아니라 전체적으로 적용되어 사용하고 있다.
- 워크숍(Workshops)은 상황에 맞게 주기적으로 실행하고 있다.
- 스탠드 업(Stand-ups)은 매일매일 10-15분 내외로 빠르게 이행하고 있다.
- 사람들은 요구사항, 사용자 스토리(user story)기능(feature), 서사시(epic) 등의 용어에 대해 명확히 이해하고 있다.
- 제품기술서는 품질항목을 기준으로 유연하게 적절한 수준으로 기술되어 있다
- 우선순위는 항상 조정된다. 이는 시간 제약 속에서 이루어진다. 시간 연장이 필요할 때는 사람을 추가 투입한다. 모든 정보는 (벽면을 활용하여) 가시적이고, 항상 최신 정보로 유지된다.
- 우선순위를 정할 때, 팀은 범위와 품질 두 가지 사항을 모두 고려한다.
- 검수조건이 모두 잘 기술된 형태로 항상 존재한다.

- 예측은 팀 기반 활동에 근거한다.
- 산출물은 린(Lean) 스타트업에 토대로 해야 하며, 각 유형의 내용을 전달하기 위해 올바른 채널이 사용되고 있다. Lean UX는 신속한 시제품 제작(rapid prototyping)과 사용자의 경험을 바탕으로 요구사항을 확정하기 위해 피드백(feedback)을 통해 제품을 디자인해서 제품의 진정한 가치(value)를 창출하는 기법이다.
- ‘디자인 씽킹’, ‘애자일 개발방법론’, ‘린 스타트업’을 토대로 한다.^[11]
- 시연(demonstration)을 자주하고 있다.

Agile 성숙도는 상기 태도측면 점검항목, 환경측면 점검항목, 프로세스측면 점검항목, 기법측면 점검항목을 고려해서 측정하게 된다.

VI. 결론

종래의 폭포수 개발방식은 소프트웨어의 비가시성(Invisibility)과 변경용이성(Changeability) 및 요구사항의 불확실성(Uncertainty) 등의 특성으로 인해 현실적으로 고객의 요구를 충족시키는 데는 어려운 것이 사실이다. 이에 반해 애자일 방식의 경험적 프로세스는 잦은 출시(Release)와 반복개발(Iteration)을 통해 소프트웨어 품질 향상과 소프트웨어를 적시에 출시하는 데 유리하다.

전통적인 개발프로세스들은 공학적으로 정형적 프로세스 제어모델을 따른다. 정형적 프로세스 제어모델은 동일한 입력에 대해서 동일한 결과가 기대 될 경우에 적합하지만, 대부분의 소프트웨어를 포함한 정보기술 개발은 경험적 프로세스 제어모델로 접근할 필요가 있다. 경험적 프로세스 제어모델은 항상 불확실성을 수반하고 포용해야 하는 특성을 갖는데, 이런 측면에서 애자일 개발 프로세스는 경험적 프로세스 제어모델로 고객을 개발팀에 참여시켜서 개발을 유도하고 관리할 수 있도록 하는 특성을 갖고 있다.

조직에 따라 프로덕트 단위로 개발할 경우 더 효율적일 수 있다. 이럴 때 부품(Component) 중심의 상호 기능팀(cross functional team) 모델이 적합할 수도 있다.

프로덕트에 따라서 빠르고 생산적인 개발기술이 필요할 경우에는 자바 개발 오픈소스 웹 프레임워크인 루비

온 레일스(Ruby on Rails)^[12] 아키텍처 디자인이나 확장성 있는 네트워크 애플리케이션(특히 서버 사이드) 개발에 사용되는 소프트웨어 플랫폼인 node.js^[13]로 그 결과물(프로덕트)을 개발하는 모델을 사용할 수도 있다.

한 조직에서 어떤 제품은 독립성이 높고, 계속해서 기능 추가가 독립적으로 가능하면 제품기반의 상호 기능팀(cross functional team)으로 독립 운영하고, 나머지는 기능단위 팀(Feature team) 모델로 운영할 수도 있다.^[14]

즉, 개발하고자 하는 서비스나 팀의 상황, 조직의 성숙도, 개발속도, 기술수준, 관리역량 등 다양한 요인에 맞춰서 팀 모델을 점차적으로 변화시켜 나가고 최적의 모델을 찾는 것이 최적의 해법이며, 애자일 기법과 툴을 잘 활용해야 한다는 결론을 얻었다.

References

- [1] Succeeding with Agile, Mike Cohn, 2010
- [2] Lean-Agile SW Development, Alan Shalloway, 2009
- [3] Kanban, David J Anderson, 2010
- [4] Succeeding with Agile, Mike Cohn, 2010
- [5] Coaching agile teams, lyssa Adkins, 2012
- [6] Takeuchi, Hirotaka; Nonaka, Ikujiro. "The New New Product Development Game" (PDF). 《Harvard Business Review》. 2010.6.9.
- [7] DeGrace, Peter; Stahl, Leslie Hulet, 《Wicked problems, righteous solutions》. Prentice Hall. ISBN 978-0-135-90126-7, 1990.10.1.
- [8] Wikimedia Foundation, Inc, "Agile Software Development", 2013.12.27.
- [9] <https://www.scrumalliance.org/community/articles/2014/april/plan-driven-versus-value-driven-planning>
- [10] AXELOS, PRINCE2 Agile™ - Agile Project Management Best Practice, 2015.
- [11] kim tae-dal, "Lean and Agile UX-based development methodology with a study on mobile design", the Korea Information & Communication Professional Engineers Association Research Journal, 2014,12
- [12] <http://rubyinstaller.rubyforge.org/>

- [13] Redis develop web applications using the Node.js, Acom Publishing, 2015
- [14] http://www.featureteams.org/feature_team_primer12.pdf

저자 소개

김 태 달(정회원)



- 1997 ~ 2015(현재) : 청운대학교 인 천캠퍼스 컴퓨터학과 정교수
- 1979 : 숭실대학교 전자계산학과(현, 컴퓨터학과)를 졸업
- 1997 : 숭실대학교 대학원에서 소프트웨어공학 전공, 석·박사학위 취득
- 1986 : 국가기술자격고시 정보처리기술사(전자계산기조직응용, 현, 컴퓨터시스템응용기술사 자격취득)
- 1997 : 정보통신기술 수석공인감리인 자격취득
- 2004 : 제17회 정보문화의 달, 국가정보화 유공자로 선정, 국무총리상 수상
- 2013 : 한국기술사회 전국기술사대회, 기술사 대상인 덕원기술사상 수상
- 2009 ~ 2013 : 한국정보통신기술사협회 제8대,9대 회장 역임
<주관심분야 : 소프트웨어 엔지니어링(Software Engineering), 정보시스템 감리(Information Systems Audit), 지능형교통시스템(ITS), 지리정보시스템(GIS), 유-헬스케어 'U-Healthcare), 독창적 전장감시 체계(logistics support, and Modernization of Command, Control, Communications, Computers, Combat and Intelligence (C5I Modernization))>