



스크럼의 이론과 실천에 대한 가벼운 안내서 Version 2.0

Pete Deemer

GoodAgile

www.goodagile.com

Gabrielle Benefield

Evolve

www.evolvebeyond.com

Craig Larman

www.craiglarman.com

Bas Vodde

Odd-e

www.odd-e.com



독자에게 알림: 스크럼에 대한 간략한 설명들은 온라인을 통해서 많이 접할 수 있으므로, 이 지침서는 스크럼을 실습하기 위한 세부적인 내용을 제공하는 것을 목표로 하고 있습니다. 이 지침서는 최종 단계의 스크럼 교육을 위해서 작성된 것이 아닙니다. 즉, 스크럼을 적용하는 것을 고려하고 있는 팀들은 Ken Schwaber의 *Agile Project Management with Scrum*이나 Mike Cohn의 *Succeeding with Agile*로 준비하고, 훌륭한 여러 스크럼 훈련방법이나 지도방법을 사용할 것을 권장합니다. 세부 사항은 scrumalliance.org에 있습니다. 조언을 주신 Ken Schwaber, Dr. Jeff Sutherland, Mike Cohn에게 감사의 말씀을 전합니다.

© 2012 Pete Deemer, Gabrielle Benefield, Craig Larman, Bas Vodde

번역: 김준현 / jheon90.kim@samsung.com

번역 및 글씨 효과는 원문을 최대한 따르며 작성했습니다.

전통적인 개발방법을 넘어서

전통적인 개발방식에서 팀은 단일 기능 조직(single-function group)으로 구성되고, 피드백은 늦게 오거나 부실하다. 또한 예측을 하기 위해서 앞부분의 계획단계에 집중하며, 분석에서부터 테스트까지 순차적인 흐름을 가진다. 이러한 전통적인 개발방식은 급변하는 오늘날의 세계에 적합하지 않다. 이런 방식은 피드백이 오거나, 학습하거나, 또는 잠재적인 투자 수익(ROI)이 발생하는 것을 지체시킨다. 왜냐하면 이 방식에서는 프로젝트의 후반부까지 실제로 동작하는 소프트웨어가 없으며 프로젝트가 투명하게 보이지가 않기 때문이다. 뿐만 아니라 개선하기 위한 능력이 부족하고, 유연함이 줄어들며, 사업적으로 그리고 기술적으로 리스크도 증가하기 때문이다.

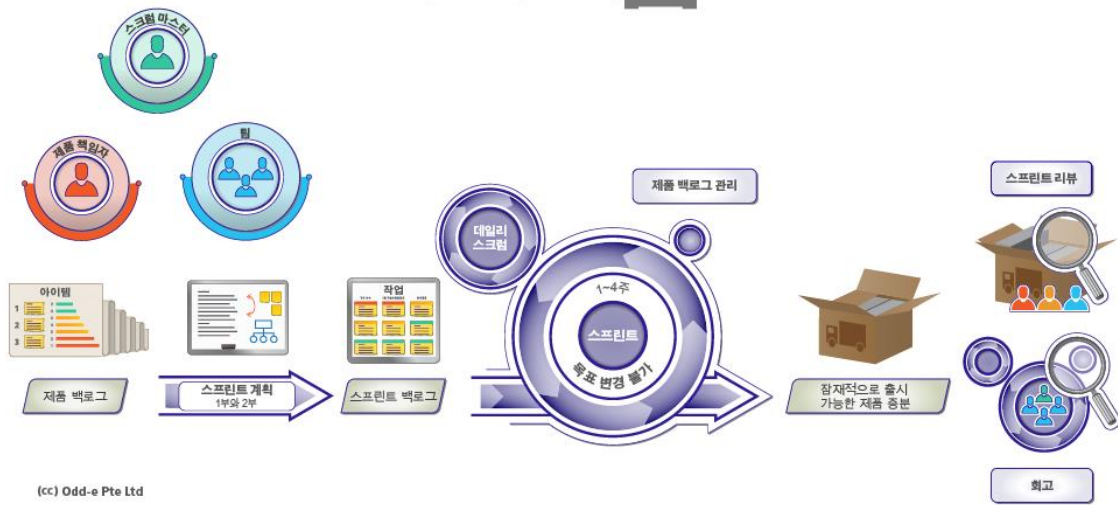
이런 방식의 대안으로는 반복적인 개발을 하는 다기능팀(CFT)이 있는데 이것 또한 오랫동안 존재하긴 했지만 전통적인 방식만큼 널리 사용되고 있지 않다.

스크럼은 단순한 프레임워크 안에서 입증된 제품 개발이라는 개념을 제시하며, 이 프레임워크는 진정한 팀(real team), 다기능팀, 자율 관리 팀, 짧고 반복적인 완전한 주기의 피드백, 변화하는데 드는 비용의 감소를 포함한다. 이런 것들을 통해서 더 민첩해 지고 피드백을 많이 받을 수 있으며 투자 수익을 더 빨리 받을 수 있게 하며 리스크를 줄일 수 있다.

개요(Overview)

스크럼은 다기능팀이 제품이나 프로젝트를 반복적이고 점진적인 방식으로 개발하기 위한 개발 프레임워크다. 여기서 개발은 스프린트라고 불리는 주기로 구성된다. 여기서 반복 주기는 각각 4주를 넘지 않으며(보통 2주), 멈추지 않고 하나가 끝나면 다음 것이 시작된다. 스프린트는 *시간이 고정(timeboxed)*되어있다. 스프린트는 일이 끝났던, 끝나지 않았던 기간이 *절대 늘어나지 않고* 정해진 날짜에 끝난다. 보통 스크럼 팀은 스프린트의 길이를 정해놓고 개선해서 주기를 더 짧게 할 수 있을 때 까지는 모든 스프린트를 그 길이로 진행한다. 매 스프린트가 시작될 때, *다기능팀(약 7명)*은 우선순위가 매겨진 리스트에서 *아이템(고객의 요구사항)*을 고른다. 팀은 스프린트가 끝날 때 어떤 결과물을 낼 수 있을 것인지에 대해서 합의를 본다. 여기서 말하는 결과물은 분명하며 정말로 “완료”라고 할 수 있는 것을 말한다. 스프린트 동안은 어떤 새로운 아이템도 추가되지 않는다. 스크럼에서는 *다음* 스프린트에 대한 변화는 수용하지만 현재의 짧은 스프린트는 작고 분명하며 상대적으로 안정적인 목표에 집중해야 한다. 매일 팀은 진척 사항에 대해서 간단하게 점검을 하고 남은 일을 완료하기 위해서 필요한 다음 단계를 조정한다. 스프린트가 끝날 때, 팀은 이해관계자와 해당 스프린트에 대해서 검토하고 무엇을 만들었는지를 보여준다. 사람들은 다음 스프린트에 반영될 수 있는 피드백을 받는다. 스크럼에서는 스프린트가 끝날 때 작동할 수 있는 제품을 만드는 것이 강조되는데, 여기서의 제품은 진짜 “완료”라고 할 수 있는 것을 뜻한다. 즉, 소프트웨어의 경우는 시스템이 통합되고 완전히 테스트가 되었으며 최종 사용자를 위한 문서가 작성되었고 잠재적으로 출시 가능한 제품을 뜻한다. 스크럼에서의 핵심 역할, 산출물, 각 활동이 그림 1에 요약되어 있다.

스크럼



(cc) Odd-e Pte Ltd

그림 1. 스크럼 개요

스크럼의 주요한 주제는 “검토(inspect)와 조정(adapt)”이다. 개발은 필연적으로 학습과 혁신 그리고 예상치 못한 것들을 포함하기 때문에 스크럼은 개발 단계를 짧게 하며, 결과물과 현재 진행 방식의 효율을 점검하고, 제품의 목표와 프로세스의 실행 방법을 조정하는 것을 중요시한다. *이것은 영원히 반복된다.*

스크럼에서의 역할들(Scrum Roles)

스크럼에는 제품 책임자(Product Owner), 팀(Team), 스크럼마스터(ScrumMaster) 라는 세 가지 역할이 있다. 이것들은 다 함께 스크럼 팀이라고 한다.

제품 책임자는 투자 수익률(return on investment, ROI)을 최대화 해야 할 책임이 있다. 그러기 위해서 제품 책임자는 기능(feature)을 식별하고 이것을 반영해서 리스트의 우선순위를 짜고, 다음 스프린트에 어떤 것이 리스트의 가장 위에 와야 하는지 결정하고, 지속적으로 리스트의 우선순위를 다시 짜고 개선한다. 제품이 상업적인 경우, 제품 책임자는 제품의 손익에 대한 책임이 있다. 내부적으로 활용되는 제품의 경우는 제품 책임자에게 제품이 상업적일 때와 같은 투자 수익률에 대한 책임은 없다. 그러나 각 스프린트마다 가장 높은 가치를 주는 아이템을 선택해야 한다는 면에서는 투자 수익률을 최대화 하는 책임이 여전히 있다. 실제로는 ‘가치’라는 단어는 불분명하다. 그리고 우선순위를 매기는 일은 주요 고객을 만족시키고자 하고, 전략적인 목적에 맞추며, 리스크에 대처하고, 개선하기 위한 활동 등의 영향을 받을 수도 있다. 어떤 경우는 제품 책임자와 고객이 같은 사람이다. 이런 경우는 내부적으로 활용되는 제품의 경우 흔히 나타난다. 다른 경우로는 고객이 다양한 니즈를 가진 수 백만의 사람일 수도 있다. 이런 경우 제품 책임자의 역할은 많은 제품 조직들에서의 제품 관리자(Product Manager)나 제품 마케팅 관리자(Product Marketing Manager)의 역할과 비슷하다. 하지만, 제품 책임자는 어쨌던 기존의 제품 관리자와는 다르다. 왜냐하면 제품 책임자는 프로젝트

관리자에게 개발에 대한 결정권을 위임하는 것이 아니라 적극적으로 자주 팀과 소통하고, 모든 이해관계자들과 함께 일하고 매 스프린트의 결과를 검토하며 우선순위를 정하기 때문이다. 중요한 것은, 스크럼에서 최종 권한을 가진 제품 책임자는 오직 한 사람이며, 제품 책임자는 업무의 가치에 대한 책임이 있다는 것이다. 제품 책임자가 혼자 일 할 필요가 없더라도 한 사람이어야 한다.

팀(개발팀이라고도 불림)은 제품 책임자가 지시하는 이를 떼면 어플리케이션이나 웹사이트 같은 제품을 만든다. 스크럼에서 팀은 “다기능적(cross-functional)”인데 이는 팀이 잠재적으로 출시 가능한 제품을 매 스프린트마다 만들기 위해서 필요한 모든 전문지식을 가지고 있어야 한다는 뜻이다. 그리고 팀은 높은 수준의 자율성과 책임을 가진 “자기조직(self-organizing, self-managing)”이어야 한다. 팀은 제품 책임자가 제안한 것들 중 한 스프린트 동안 몇 개의 아이টে임을 완성할 지를 결정해야 하며 그 목표를 달성하기 위해서 어떤 방법이 최선인지도 결정해야 한다.

각 팀원들은 그저 *팀원*이다. 스크럼을 적용한 그룹에서는 고정된 전문가(specialist)라는 타이틀은 없다. 그 말은 사업 분석가, DBA, 아키텍트, 팀의 리더, 인터랙션/UX 디자이너, 프로그래머 같은 것들이 없다는 뜻이다. 매 스프린트가 진행되는 동안 팀이 스스로 정한 목표를 달성하기 위해서 팀원들은 어떤 방식으로든 함께 일한다.

오직 팀원만 있기 때문에 팀은 다기능적일 뿐만 아니라 *다중학습(multi-learning)*이라는 측면도 보여준다. 모든 사람들은 확실히 특별한 장점을 가지고 있지만 계속해서 다른 전문 지식을 배운다. 모든 사람들은 1차, 2차 그리고 심지어 3차 기술을 가진다. 이는 “일이 있는 곳은 어디든 간다”는 것을 뜻한다. 즉, 개개인들은 아이টে이들이 완료되도록 돕기 위해서 다소 덜 친숙한 분야의 일도 맡는다. 예를 들면 1차 기술이 인터랙션 디자인인 사람이 2차 기술로 테스트 자동화를 가질 수도 있다. 또는 누군가 1차 기술이 기술문서 작성인 사람이 분석이나 프로그래밍 또한 도와줄 수도 있다는 뜻이다.

스크럼에서의 팀은 7명에서 2명 안팎이다. 그리고 소프트웨어 제품의 경우 팀에는 분석, 개발, 테스트, 인터페이스 디자인, 데이터베이스 디자인, 아키텍처, 문서작성 등의 기술을 가진 사람들이 있어야 할 것이다. 팀은 제품을 개발하고 제품 책임자에게 어떻게 제품을 훌륭하게 만들 것인지에 대한 아이디어를 준다. 스크럼에서의 팀은 모든 팀원들이 스프린트 동안 하나의 제품을 위해서 100% 몰두해서 일할 수 있을 때 가장 생산적이고 효율적이다. 즉, 팀은 주의가 분산되거나 흐름이 바뀌면서 낭비되는 것을 막기 위해서 여러 제품이나 프로젝트에 걸쳐져서 동시에 여러 작업을 하게 되는 것을 피해야 한다. 팀이 안정적인 것은 높은 생산성과 관련이 있으므로 팀원이 바뀌는 것을 피해야 한다. 많은 사람들로 구성된 제품 그룹은 여러 팀들로 구성되는데 각 팀들은 그들의 일에 대해서 긴밀하게 조율하며 제품의 서로 다른 기능(feature)들에 집중한다. 하나의 팀이 고객 중심의 기능들을 완성하기 위한 모든 일(계획, 분석, 프로그래밍, 테스트)을 종종 하므로 이런 팀은 또한 *기능팀(feature team)*이라고도 불린다.

스크럼마스터는 제품 그룹이 사업적인 가치를 만들기 위해서 스크럼을 배우고 적용하는 것을 돕는다. 스크럼마스터는 팀과 제품 책임자 그리고 조직이 성공할 수 있기 위해서 가능하다면 어떤 것이든 돕는다. 스크럼마스터는 팀원들의 관리자가 아니며 프로젝트 관리자도 아니고 팀 리더도 아니며 팀 대표자도 아니다. 대신 스크럼마스터는 팀에게 도움을 준다. 즉, 스크럼마스터는 장애물을 제거하고 팀을 외부의 간섭으로부터 보호하며 팀이 새로운 개발 방식에 적응할 수 있도록 돕는다. 스크럼마스터는 제품 책임자와 팀 그리고 나머지 조직원들이 스크럼을 능숙하게 사용할 수 있도록

가르치고 지도하고 설명해준다. 스크럼마스터는 코치이자 선생님이다. 스크럼마스터는 제품 책임자나 관리를 하는 사람들을 포함한 모두가 스크럼의 원칙과 실천 방법들을 반드시 이해하여 조직이 애자일 개발을 성공적으로 마치기 위해서 필요한 어려운 변화들을 극복할 수 있게 돕는다. 스크럼은 팀이나 제품 책임자의 효율에 대한 장애물과 위협이 눈에 보이도록 만든다. 그러므로 이런 문제들을 해결할 수 있도록 활발하게 일할 팀에 소속된 스크럼마스터가 있는 것이 중요하다. 그렇지 않다면 팀이나 제품 책임자는 성공하기 어렵다는 것을 깨닫게 될 것이다. 비록 작은 팀의 경우는 팀원이 기존에 하던 일을 줄이고 이 역할을 할 수도 있지만, 그래도 모든 시간 동안 이 일에만 전념하는 스크럼마스터가 있어야 한다. 훌륭한 스크럼마스터는 공학, 디자인, 테스트, 제품 관리, 프로젝트 관리, 품질 관리 등 어떤 배경이나 지식 분야에서도 나올 수 있다.

스크럼마스터와 제품 책임자는 초점이 매우 다르며 병행할 경우 혼란이나 충돌이 생기기 때문에 같은 사람이 맡을 수 없다. 두 역할을 병행하면 발생하게 되는 혼란 문제 중 하나는 세부사항까지 관리하는(micro-managing) 제품 책임자이다. 이는 스크럼이 요구하는 자기조직 팀과 맞지 않다. 기존의 관리자와는 다르게 스크럼마스터는 사람들에게 무엇을 해야 하는지 또는 할당된 일을 말해주지 않는다. 스크럼마스터는 프로세스를 따를 수 있게 돕고 팀이 스스로 조직하고 관리하도록 도와준다. 만약 스크럼 마스터가 이미 팀을 관리하는 위치에 있다면, 그들은 스크럼을 통해 팀이 성공하기 위해서 마음가짐이나 소통하는 방식을 크게 바꿀 필요가 있다.

참고: 스크럼에서는 프로젝트 관리자의 역할이 전혀 없다. 왜냐하면 전혀 필요하지 않기 때문이다. 기존의 프로젝트 매니저가 맡았던 일들은 스크럼의 세 가지 역할에 나누어 할당되는데 스크럼 마스터보다는 팀과 제품 책임자에게 대부분 할당한다. 스크럼을 하면서 프로젝트 관리자를 추가한다는 것은 스크럼을 근본적으로 잘못 이해하고 있다는 것을 말한다. 그리고 이런 경우 일반적으로 책임들 간의 충돌, 분명하지 않은 권한, 차선의 결과 같은 것들이 발생한다. 때론 프로젝트 관리자가 스크럼마스터의 역할을 하게 될 수 있지만 이런 접근방식이 전적으로 성공할 수 있을지는 개인이 얼마나 두 역할의 근본적인 차이에 대해서 이해를 했는지, 그날그날의 역할과 성공하기 위한 마음가짐을 모두 이해했는지에 달려있다. 스크럼마스터의 역할을 완전히 이해하고 성공하기 위한 핵심 기술들을 개발하기 위한 좋은 방법은 Scrum Alliance의 Certified ScrumMaster 교육을 받는 것이다.

이 세 가지 역할 뿐만 아니라 관리자, 고객, 최종 사용자 같은 제품의 성공에 기여하는 다른 이해관계자들도 있다. 직무 관리자(functional manager)나 공학 관리자(engineering manager)와 같은 이해관계자들은 새로운 역할을 찾을 수도 있다. 하지만 그 역할이 여전히 가치가 있을지라도 스크럼을 적용할 때는 변해야 한다. 예를 들면,

- 그들은 스크럼의 규칙들과 참 뜻을 존중함으로써 팀을 지원한다.
- 그들은 팀과 제품 책임자가 찾은 장애물을 제거하는 것을 돕는다.
- 그들은 그들의 전문 지식과 경험을 쓸 수 있도록 한다.

스크럼에서 이런 사람들은 이전에 “유모” 역할(업무를 할당하고 현황 보고서를 받고 다른 형태의 세세한 관리를 함)을 하며 보냈던 시간들을 팀의 “구루”나 “하인”의 역할(멘토링, 코칭, 장애물 제거를 돕기, 문제를 해결하는 것을 도와주기, 창의적인 조언 제공, 팀원들의 기술 개발 지도)을 할 시간으로

바뀌어야 한다. 이 변화에서 관리자들은 그들의 관리 스타일을 바꿀 필요가 있다. 예를 들면 단순히 해결책을 정해서 팀에게 할당하는 방식이 아니라 팀이 문제를 해결법을 찾을 수 있도록 소크라테스식 문답법을 사용한다.

제품 백로그(Product Backlog)

팀이 스크럼을 시작하려면 첫 번째 스프린트를 시작하기 전에 **제품 백로그(Product Backlog)**가 있어야 한다. 제품 백로그는 사용자 중심의 기능(feature)들을 우선순위를 매겨(1, 2, 3, ...으로 매김) 나열한 리스트이다.

제품 백로그는 제품 수명 기간 동안 계속 존재하고 개선되면서 해당 제품의 로드맵 역할을 한다. (그림 2, 그림 3). 어떤 관점에서 보든, 제품 백로그에는 “팀이 해야 하는 모든 것이 우선순위에 따라 기록”된 것이 한 눈에 명확히 드러나야 한다. 하나의 제품에 대해서는 오직 하나의 제품 백로그만이 존재한다. 이 말은 제품 책임자는 팀을 포함해서 이해관계자들의 관심사를 전반적으로 반영해서 우선순위를 결정해야 한다는 것을 뜻한다.

		새로운 추정치 스프린트가 끝날 때 남은 것...							
우선순위	아이템	세부사항 (위키주소)	초기 추정치	1	2	3	4	5	6
1	구매자로서, 나는 책 한 권을 쇼핑카트에 담고 싶다 (위키 페이지에 있는 UI 스케치를 볼 것)	...	5						
2	구매자로서, 나는 쇼핑카트에 있는 책 한 권을 제거하고 싶다	...	2						
3	트랜잭션 처리 성능 향상(위키에 있는 목표 성능치 측정기준을 볼 것)	...	13						
4	신용카드 승인 속도 향상을 위한 솔루션 조사(위키에 있는 목표 성능치 측정기준을 볼 것)	...	20						
5	모든 서버를 아파치 2.2.3으로 업그레이드	...	13						
6	스크립트 에러를 처리하는 순서 진단, 결정(bugzilla ID 14823)	...	3						
7	쇼핑객으로서, 나는 위시 리스트를 만들고 저장하고 싶다	...	40						
8	쇼핑객으로서, 나는 내 위시 리스트에 아이템을 넣거나 빼고 싶다	...	20						

그림 2. 제품 백로그



그림 3. 시각적 관리: 벽에 붙어있는 제품 백로그 아이템들.

제품 백로그에는 다양한 아이템들이 포함되는데 “모든 사용자들은 장바구니에 책을 담을 수 있다”와 같은 우선적으로 고객을 위한 새로운 기능이 포함된다. 뿐만 아니라 “해당 시스템을 C++기반에서 Java기반으로 변환한다”와 같은 개발에 대한 발전 목표나 “테스트 속도를 향상시킨다”와 같은 팀에서의 개선 목표, “신용카드 인증 속도를 향상하기 위한 해결책을 조사한다”와 같은 연구 업무 또는 결함이 많지 않다면 “스크립트 오류를 처리하는 명령어를 진단하고 고친다”와 같이 발견된 결함을 포함시킬 수도 있다. (결함이 많은 시스템의 경우는 보통 별도의 결함 추적 시스템을 가진다.)

제품 백로그에 들어갈 아이템들은 어떤 방식으로든 분명하고 계속 유지될 수 있도록 표현된다. 많이 오해하고 있는 것과는 달리 제품 백로그는 사용자 스토리가 아니라 단지 아이템만 담는다. 이러한 아이템들은 사용자 스토리, 유스케이스(use cases) 또는 팀이 유용하다고 생각하는 요구사항 작성 방법으로 표현될 수 있다. 어떤 방식을 활용하든, 대부분의 아이템들은 고객에게 가치를 전달하는 것에 초점이 맞춰져야 한다.

좋은 제품 백로그는 DEEP로 표현할 수 있다.

Detailed appropriately. 최우선 순위의 아이템은 낮은 우선순위의 아이템보다 먼저 진행되기 때문에 더 상세하고 세밀하게 기술되어 있어야 한다. 예를 들면, 백로그의 상위 10%는 나머지 90%에 비해 더 작은 단위로 상세하게 분석되어 있어야 한다.

Estimated. 현재 릴리즈의 각 아이템들은 추정이 필요하며 매 스프린트마다 팀원들이 무언가를 배우고 새로운 정보가 생기므로 다시 추정되어야 한다. 팀은 제품 책임자에게 제품 백로그의 각 아이템에 해당하는 *일의 양(effort)*과 *기술적인 위험성(technical risk)*에 대한 추정치를 제공해야 한다. 제품 책임자와 다른 사업적인 이해관계자들은 제품 요구사항의 가치에 대한 정보를 다양한 이해관계자들에게 전달해야 한다. 이 정보에는 수익이나 비용 절감, 사업적인 리스크, 중요성이 포함될 수 있다.

Emergent. 학습한 내용이나 변동되는 것들에 맞추어 제품 백로그는 자주 개선된다. 매 스프린트마다 우선순위에 따라 아이템들이 추가되고 제거되고 수정되며 쪼개지고 변경된다. 그러므로 고객의 니즈 변화, 새로운 아이디어, 경쟁사의 움직임, 기술적인 걸림돌 등을 반영하기 위해서 제품 책임자는 계속해서 제품을 업데이트 해야 한다.

Prioritized. 제품 백로그의 최우선 순위 아이템부터 1에서 N까지의 순서로 우선순위가 매겨져야 한다. 최우선 순위의 아이템들은 *적은 비용으로 큰 가치를 내는 것이어야 한다(bang for your buck)*. 또 다른 방식으로는 *큰 위험을 포함하고 있는 아이템의 우선순위를 높게(tackle high risks early, before the risks attack you)* 잡는 방법이 있다.

전통적인 개발 방식에서는 높은 효과 순으로 결과를 내는 것이 강조되지는 않지만 스크럼에서 이는 주요한 개념이 된다. 그러므로 제품 책임자는 이런 “사업적 가치”를 가늠하는 법을 배워야 하며 스크럼마스터는 이를 도와야 한다. 그렇다면 “사업적 가치”는 무엇일까? 어떤 제품 팀들은 수익이나 절감된 비용, 이해관계자의 선호도 또는 시장 차별화 등의 요소 등으로 어렵짐작한 것들을 종합해서 단순하게 상대적인 점수를 계산한다. 또 다른 방법으로는 만약 고객이 특정 아이템에 대해서 비용을 지불한다면, 그 아이템의 수익을 가치로 생각할 수 있다. 이러한 아이템 위주의 추정 방법을

적용하기에 불명확할 경우, 전반적인 경영 성과를 바탕으로 접근할 수 있다. 이런 경우는 성과에 관련된 여러 개의 아이템들이 한 번에 전달돼야 가치를 만드는 경우가 해당되며 이런 경우 제품 책임자는 최소 기능 제품(Minimal Viable Product)에 대해서 다음 번에 만들 증분(increment)을 정의한다.

일의 양을 추정하는데 흔히 쓰이는 기술 중 하나는 일의 양, 복잡성, 불확실성 등을 고려해서 “스토리 포인트”나 단순히 “포인트” 단위로 상대적인 크기를 추정하는 것이다.

이러한 방법들은 단지 제안일 뿐이다. 스크럼에서는 제품 백로그의 아이템들을 표현하거나 우선순위를 매기는 방법에 대해서 정의하지 않고 추정 기술이나 추정 단위에 대해서도 고려하지 않는다.

스크럼에는 매 스프린트마다 일이 얼마나 끝났는지를 추적하는 기술이 있다. 예를 들어 스프린트마다 평균적으로 26포인트가 완료된다고 가정해 보자. 만약 이 평균이 계속 유지되고 다른 변화가 없으면 모든 기능(feature) 개발이 완료되는 릴리즈 날짜는 언제인지, 정해진 기간 내에 얼마나 많은 기능 개발을 완료할 수 있는지 계획할 수 있다. 이러한 평균값을 “속도(velocity)”라고 하며, 속도는 제품 백로그의 아이템을 추정할 때와 같은 단위를 사용한다.

제품 백로그의 아이템들은 완료를 위해 필요한 일의 양에 따라서 변할 수도 있다. 제품 백로그를 개선하기 위한 워크샵이나 스프린트 계획 회의 때 큰 아이템을 작게 쪼개거나 작은 아이템을 통합한다. 다가오는 스프린트의 제품 백로그 아이템들은 팀이 이 아이템들에 대해서 이해할 수 있을 만큼 충분히 세분화되어서 스프린트 계획 때 팀이 유의미한 예측을 할 수 있어야 한다. 이런 세분화된 크기를 “실행가능한(actionable)” 크기라고 한다.

시간과 비용이 많이 드는 주요한 기술 개선사항의 경우는 제품 백로그에 포함되어야 한다. 왜냐하면 이런 개선사항은 사업 투자가 될 수도 있고, 궁극적으로 이 투자를 할지는 사업 지향적인 제품 책임자가 결정할 것이기 때문이다. 참고로 스크럼에서, 팀에게는 스프린트 동안 제품 백로그에 있는 아이템 중 몇 개를 할지를 정할 수 있는 독자적인 권한이 있다. 그러므로 팀원들은 가벼운 기술적 개선사항에 대해서는 만약 그런 개선사항들이 사업을 진행하면서 필요한 정상적인 것들이고 개발자들이 일을 제대로 하기 위해서 필요한 것이라면 독자적으로 맡아서 처리할 수 있다. 물론 그렇긴 하더라도 매 스프린트마다 팀은 대부분의 시간을 내부의 기술적인 업무 보다는 제품 책임자가 정한 목표를 위해서 쓴다.

스크럼에 대한 오해 중 하나는 스크럼을 하면 구체적인 명세서(specification)를 쓰지 않아도 된다는 것이다. 실제로 얼마만큼의 세부적인 내용들이 필요한지는 팀이 결정하는 것이고, 이런 결정은 백로그 아이템마다 팀의 이해도(insight)같은 요인들에 따라 다르다. 백로그에는 무엇이 중요한지를 최소한으로 명시해야 한다. 즉, 모든 세부사항을 기록하기 보다는 해당 아이템을 이해하는데 필수적으로 필요한 것이 무엇인지 명확하게 기록하고 팀, 제품 책임자 그리고 이해관계자들과의 대화를 통해 이를 늘려나가야 한다. 우선순위가 낮은 제품 백로그의 아이템은 한동안 진행되지 않을 것이고 대체로 크기가 크며 상세화가 덜 되어있다. 반면 우선순위가 높은 제품 백로그의 아이템은 빠른 시일 내에 진행될 예정이며 더 상세화가 되어있다.

완료 기준(Definition of Done)

공식적으로 매 스프린트의 산출물은 잠재적으로 출시 가능한 제품 증분(Potentially Shippable Product Increment)이라고 부른다. 첫 번째 스프린트를 시작하기 전에 제품 책임자, 팀, 스크럼 마스터는 제품 백로그의 아이템이 잠재적으로 출시 가능하기 위해서 필요한 모든 것에 대해 검토해야 한다. 제품을 출시하기 위해 필요한 모든 활동은 잠재적으로 출시 가능이라는 말의 정의에 포함되어야 하며 그렇기 때문에 스프린트 동안 끝나야 한다.

안타깝게도 팀이 스크럼을 적용하기 시작하면 종종 매 스프린트마다 잠재적으로 출시 가능한 제품을 전달하고자 하는 목표를 이루지 못한다. 이런 일은 팀이 자동화가 덜 되었거나 충분히 다기능적이지 않기 때문이다. (예시. 기술 관련 문서작성자가 다기능팀에 아직 없음) 시간이 지나면 팀은 더 나아져서 매 스프린트마다 잠재적으로 출시 가능한 제품 증분을 전달할 수 있을 것이다. 그러나 시작을 위해서는 현재의 역량에 대한 기준을 만들 필요가 있다. 이 기준들이 완료 기준으로 기록된다.

스프린트가 시작하기 전에 제품 책임자와 팀은 완료 기준에 대해서 합의를 볼 필요가 있다. 이 활동은 잠재적으로 출시 가능한 제품 증분을 만들기 위한 활동의 일부이다. (좋은 팀의 경우도 마찬가지이다.) 팀은 이 완료기준에 맞추어 스프린트의 업무를 계획한다.

좋은 제품 책임자는 항상 완료기준이 가능하면 잠재적으로 출시할 수 있을 수준에 가깝게 잡혀서 개발이 투명해지고 *지연*되는 경우가 줄어들며 *리스크*가 감소하기를 원한다. 만약 완료기준이 잠재적으로 출시 가능한 정도가 아니라면 *리스크와 지연*이 발생하게 될 릴리즈를 하게 될 때까지 계속 작업이 지연될 것이다. 이런 지연된 작업을 때론 *완료되지 않은 작업*이라고 부른다.

스크럼 팀은 끊임없는 개선을 해야 하고 이런 개선 사항들이 완료 기준을 늘리는데 반영된다.

스프린트 계획(Sprint Planning)

요약: 스프린트를 준비하기 위한 회의로, 2개의 파트로 나뉜다. (파트1은 “무엇(what)”, 파트2는 “어떻게(how)”에 대해 다룬다.)

참가자: 파트1: 제품 책임자, 팀, 스크럼마스터. 파트2: 팀, 스크럼마스터, 제품 책임자(선택 사항이지만 질문이 있으면 연락할 수 있어야 한다.)

기간: 각 파트는 스프린트의 한 주당 한 시간 이내로 제한한다.

스프린트 계획 회의(Sprint Planning Meeting)는 매 스프린트를 시작할 때 진행되며, 2개의 회의로 나뉘어 진행되는데 첫 번째 회의를 **스프린트 계획 파트1(Sprint Planning Part One)**이라고 한다.

스프린트 계획 파트1에서는 제품 책임자와 팀이 제품 백로그 중 제품 책임자가 이번 스프린트에서 진행되길 원하는 우선순위가 높은 아이템들을 검토한다. 대체로 이 아이템들은 지난 스프린트에서 제품 백로그 개선을 하는 동안 잘 분석되었기 때문에 이 회의에서는 간단히 질문에 대해서 다시 한번 명확히 하는 작업만 진행된다. 제품 책임자와 팀은 제품 백로그의 우선순위가 높은 아이템들의 목표와 내용에 대해서 정의하며, 이런 우선순위를 통해 팀원들은 제품 책임자의 생각을 알게 된다. 파트1은 제품 책임자가 무엇을 원하고 왜 그것이 필요한지에 초점이 맞춰져 있다. 회의 후반부에 제품 책임자는 자리를 떠나도 되지만 2단계 회의에는 참여(전화도 가능) 할 수 있어야 한다.

파트1에서 팀과 제품 책임자는 **스프린트 목표(Sprint Goal)**을 정한다. 이것은 스프린트의 목표를 담은

요약된 문장이며 여기서의 목표는 이상적으로는 짜임새가 있는 주제를 담고 있어야 한다. 팀은 또한 실제로 어떤 결과물을 만들어 낼 수 있을지를 고려해서 스프린트 목표의 범위를 유연하게 정할 수 있다. 왜냐하면 팀은 스프린트의 시간적 제약 때문에 몇몇 아이템을 제외해야 함에도 불구하고 해당 스프린트의 목표에 맞는 실질적이고 “완성된” 무언가를 내놓아야 하기 때문이다.

그렇다면 스프린트에 진행되는 아이템의 규모는 어느 정도가 적당할까? 각 아이템들은 전체 스프린트에 비해서 현저히 작게 쪼개져야 한다. 일반적인 지침으로는 각각의 아이템들은 스프린트 기간의 1/4 이내에 완수될 수 있을 정도의 크기로 추정하는 것이 좋다.

스프린트 계획 파트2에서는 팀이 하기로 결정한 아이템을 “어떻게(how)” 수행할 지에 대해서 초점을 맞춘다. 팀은 제품 백로그의 우선순위가 높은 것부터 순차적으로 해당 스프린트가 끝날 때까지 할 수 있는 아이템의 수를 예측한다. *이것이 스크럼에서 핵심이다. 스크럼에서는 완수해야 할 작업을 제품 책임자가 팀에 할당하는 것이 아니라, 팀이 스스로 할 수 있는 작업의 양을 결정하는 것을 중요시한다.* 팀이 나름대로 분석과 계획을 한 것을 토대로 작업의 양이 예측되기 때문에 그 결과를 더욱 신뢰할 수 있다. 제품 책임자는 팀이 얼마나 많은 양의 작업을 할지 결정하지는 않지만, 팀이 하기로 한 아이템들이 제품 백로그에서 우선순위가 높은 것부터, 즉 가장 중요하다고 생각하는 것부터 뽑아져 나왔다는 것을 알 수 있다. 팀은 우선순위가 아래에 있는 제품에 대해서도 영향력을 행사할 수 있다. 이는 보통 팀과 제품 책임자 둘 다 현재 우선순위가 아래에 있는 아이템이 더 위에 있는 것이 적절하다고 생각하는 경우이다.

스프린트 계획 회의는 작업을 완수하기 위한 진지한 예측 작업이고 팀은 여기에 신중하게 임한다. 이런 이유로 때때로 회의가 몇 시간씩 진행되기도 하는데, 스프린트 계획 회의는 2주 단위의 스프린트를 기준으로 4시간을 넘지 않도록 해야 한다. 파트1, 파트2 모두 같은 시간제한을 적용하여 2주 단위의 스프린트를 기준으로 각 파트가 2시간을 넘지 않도록 해야 한다.

스크럼은 스프린트 계획 파트2를 정확히 어떻게 해야 하는지에 대해서 정의하지는 않는다. 어떤 팀은 스프린트에 진행 할 목표치를 정하는데 이전 스프린트의 속도를 사용하기도 하고, 어떤 팀은 그들의 수용능력을 처음으로 계산해보기 위해서 더 세분화된 방법을 사용하기도 한다.

수용능력을 활용하는 방식으로 목표치를 정하고자 한다면, 팀은 스프린트 파트2를 진행하면서 각 팀원들이 스프린트와 관련된 업무를 수행하는데 얼마만큼의 시간을 쓸지를 추정한다. 스프린트와 관련된 업무를 수행하는데 쓸 시간이라는 것은 회의를 참석하거나 메일을 보내거나 휴식하는 시간은 제외한 개념이다. 대부분의 사람들의 경우는 하루에 4~6시간 정도를 여기에 쓸 수 있으며 이것을 이번 스프린트에 대한 수용능력으로 생각할 수 있다. 수용능력이 정해지고 나면 해당 기간 동안 제품 백로그에 있는 아이템 중 몇 개를 할지, 또 어떻게 이것들을 완수할 것인지를 생각해 보아야 한다. 이 과정은 보통 화이트보드로 디자인에 대한 토론을 하면서 시작된다. 팀원들이 전반적인 디자인에 대해서 이해하고 나면 제품 백로그에 있는 아이템들을 작업으로 세분화한다. 제품 백로그에 있는 아이템을 가져오기 전에 팀원들은 지난번 스프린트의 회고 때 나온 개선 목표를 위한 작업을 만드는데 집중할 수도 있다. 그리고 나면 팀은 제품 백로그에서 제품 책임자가 가장 우선순위를 높게 설정해놓은 아이템을 골라서 ‘충분’할 때까지 세분화 시키는 작업을 한다. 각 아이템에 대해서 세분화된 작업들을 리스트에 포함시킬 수도 있고 제품 백로그의 아이템을 완료하는데 걸리는 시간이 2시간 정도로 규모가 작다면 그대로 포함시킬 수도 있다. 스프린트 동안 해야 하는 일에 대한 이

리스트를 스프린트 백로그(Sprint Backlog)라고 한다. (그림 4, 그림 5)

제품 백로그 아이템	스프린트 작업	지원자	초기 업무량 추정치	업무의 양에 대한 새로운 추정치 하루가 끝날 때 남은 것...					
				1	2	3	4	5	6
구매자로서, 나는 쇼핑카트에 책 한 권을 넣고 싶다.	데이터베이스 수정		5						
	웹페이지 생성(UI)		8						
	웹페이지 생성(자바스크립트 로직)		13						
	자동화된 인수 시험 작성		13						
	구매자 도움 웹페이지 업데이트		3						
...									
트랜잭션 처리 성능 향상	DCP 코드 병합과 레이어 수준 테스트 완료		5						
	자동화된 인수 시험 작성		8						
	구매자 도움 웹페이지 업데이트		13						

그림 4. 스프린트 백로그를 만드는 방식의 한 예시

스프린트 계획 회의가 끝날 즈음, 팀은 스프린트가 끝날 때 어떤 결과물을 내놓을 것인지에 대한 현실적인 목표를 정한다. 일반적으로 이것을 스프린트 약속(Sprint Commitment)이라고 하는데 이는 팀이 목표를 달성하기 위해서 최선을 다할 것을 약속하는 것을 의미한다. 안타깝게도 이 단어는 때론 “열심히 한다(Going for it)”는 의미가 아니라 피로 쓴 맹세라는 오해를 받는다. 이런 오해가 생기지 않도록 스프린트의 목표를 제품 책임자에게 전달할 때는 “예측”이라는 단어로 표현한다.

스크럼은 가령 테스트만 하는 “테스터”와 같이 “직책에 맞는 일”만 하기 보다는 여러 기술(multi-skilled)을 가지기를 권장한다. 즉, 팀원들은 “일이 있는 곳으로 가서” 할 수 있는한 도와준다. 만약 테스트 업무가 많다면 모든 팀원들이 도와줄 것이다. 이 말이 모든 사람이 제너럴리스트(generalist)가 되어야 한다는 의미는 아니다. 물론 어떤 사람들은 특별히 테스트나 다른 기술이 있겠지만 팀원들은 함께 일하고 새로운 기술을 서로 배워야한다. 결과적으로 사람들은 스프린트 계획에서 업무를 만들거나 추정할 때 “최선을 다 할 수 있는” 모든 일에 자원할 필요가 없으며 그런 것이 적절하지도 않다. 그보다는 새로운 업무를 고를 때는 한 번에 하나의 업무에만 자원하고, 업무를 고를 때는 학습(아마 짝으로 일하게 될 스페셜리스트를 통한)이라는 목적을 고려하는 것이 더 좋다. 이런 이유로 스프린트 계획 단계에서 업무를 할당하는 것이 아니라, 스프린트가 진행되는 동안 ‘필요에 따라’ 업무를 할당한다.

존은 특정한 업무를 할 때 앞서 말한 것들은 거의 하지 못한다. 왜냐하면 다른 사람을 가르치는 것은 너무 시간이 오래 걸리거나 불가능하기 때문이다. 아마도 존은 팀에서 그림을 그리는 예술적인 기술을 가진 유일한 사람일 것이다. 다른 팀원들은 만약 직장에서의 수명이 이 업무에 달려있다면 “막대기 같은 사람”을 그리지는 못할 것이다. 만약 이런 일이 흔히 일어난다면 뭔가 잘못된 것이다. 이런 일이 일어나는 경우에는, 계획되어있는 존이 완료해야만 하는 그림을 그리는 작업 전체가 짧은 스프린트 동안 가능한지에 대해서 생각해봐야 한다.

많은 팀들은 벽 크기의 업무 보드 형식의 스프린트 백로그를 만드는데 이는 보통 스크럼 보드(Scrum Board)라고 불린다. 이 보드에 있는 포스트잇에 적힌 작업들은 스프린트 기간 동안 그림 5와 같이 “할 일(To Do)”, “작업 중(Work In Progress)”, “완료(Done)”라고 적힌 열 사이를 이동한다.



Figure 5. 시각적인 관리 - 벽에 붙여 있는 스프린트 백로그 업무들

스크럼의 기본적인 하나의 특징은 팀이 스프린트의 목표를 한 번 설정하면, 모든 추가사항이나 변경사항들은 다음 스프린트까지 미뤄져야 한다는 것이다. 이 말은 만일 제품 책임자가 스프린트 중에 팀이 했으면 하는 새로운 아이템을 결정해도 다음 스프린트가 시작할 때 까지는 바꿀 수 없다는 뜻이다. 만약 외부 상황에 의해서 우선순위가 많이 바뀌거나 이 일을 계속 했을 경우 팀이 시간을 낭비하는 꼴이 된다면 제품 책임자나 팀은 스프린트를 중단할 수 있다. 팀이 스프린트를 중단하면 새로운 스프린트 계획 회의를 통해서 새로운 스프린트를 시작한다. 보통 이런 일이 일어나는 것을 막는 것이 좋다. 왜냐하면 이런 경우 제품 책임자와 팀이 사기가 하락해서 극단적인 결정에 의지하도록 하기 때문이다.

스프린트 동안 목표가 바뀌지 않도록 팀을 보호하는 것은 강력하고 긍정적인 영향을 준다. 첫 번째로, 팀은 목표가 바뀌지 않을 것임을 확실하게 알고 일을 하기 때문에 확실하게 완료하는데 초점을 더 둘 수 있다. 두 번째로 제품 책임자는 스프린트에 자신이 제품 백로그에 우선순위를 매기고 팀에게 준 아이템들에 대해서 제대로 생각해 보게 된다.

이런 스크럼의 규칙들을 따르면서 제품 책임자는 두 가지를 얻는다. 첫 번째로 제품 책임자는 팀이 실질적이고 분명한 일들을 끝내기 위해서 최선을 다 한다는 것을 알게 된 것에 확신할 수 있다. 시간이 지나면서 팀은 실질적인 예측으로 일을 고르고 끝내는데 꽤 능숙해 질 수 있다. 두 번째로 제품 책임자는 다음 스프린트가 시작하기 전에는 어떤 변경이든 제품 백로그에 반영할 수 있다. 이런 측면에서 일을 추가하거나, 빼거나, 수정하거나, 우선순위를 다시 정하는 것이 모두 가능하고 수용된다. 제품 책임자는 현재 스프린트가 진행되는 동안 개발 중인 아이템에 대해서는 변경을 할 수 없지만 원하는 변경사항을 반영하기 위해서 한 스프린트 기간 이하만 기다리면 된다. 방향의 변화, 요구사항의 변화, 평범한 마음가짐의 변화 같은 변화를 둘러싼 오명은 사라졌고 이런 이유들로 제품 책임자들은 보통 스크럼에 대해서 누구 못지않게 열정적이다.

데일리 스크럼(Daily Scrum)

요약: 팀원들 사이의 최근 정보 공유와 협력

참가자: 팀은 필수, 제품 책임자의 참가는 선택사항, 스크럼마스터는 보통 참가하지만 팀에도 이 역할을 할 사람이 한 명 있도록 해야 함

기간: 최대 15분 이내

스프린트가 시작되면 팀은 하나의 스크럼의 주요 활동을 하게 되는데 이것이 **데일리 스크럼(Daily Scrum)**이다. 데일리 스크럼은 매일 정해진 시간에 15분 이하로 하는 짧은 회의이다. 모든 팀원은 여기에 참석해야 한다. 간단하게 덧붙이자면 이 회의는 모두가 서서 진행하기를 권장한다. 이 회의에서 팀원들은 각자가 한 업무에 대해서 서로 맞추고 장애 요소에 대해서 말한다. 데일리 스크럼에서 각 팀원들은 한 명씩 세 가지를 *다른 팀원들에게* 말한다. (1) 지난 회의 이후로 무엇을 완료했는가? (2) 다음 회의 전에 무엇을 완료할 것인가? (3) 이것을 하는데 장애 요소는 무엇인가? 참고로 데일리 스크럼은 관리자에게 상태를 보고하는 회의가 아니다. 이 회의는 자율적인 조직이 무엇이 진행되고 있는지에 대해서 서로 공유하고 협력하는 시간이다. 누군가가 장애 요소에 대해서 말하면 스크럼 마스터는 팀원들이 그것을 해결하도록 도와주는 역할을 한다. 데일리 스크럼 동안은 깊이가 얕거나 없는 토론만 하며 그 주제는 세 가지 질문에 대한 답과 관련된 것이다. 만약 논의가 필요하다면 데일리 스크럼이 끝난 후 바로 이어서 한 개 이상의 유사한 회의를 하며, 스크럼의 누구도 꼭 참여할 필요는 없다. 이런 연이은 회의를 하는 경우는 흔히 있으며 여기서 일부 또는 전체 팀원들은 데일리 스크럼 때 들은 정보에 대한 조정을 한다. 다른 말로 하면 이것은 또 다른 검토과 조정(inspect and adapt)의 반복이다. 팀이 스크럼을 처음 한다면 일반적으로 관리자나 또는 권한이 있다고 여겨지는 사람들이 데일리 스크럼에 들어오지 *않기*를 권장한다. 이런 위험한 행동을 통해 팀은 “감시”당하고 있다고 생각하게 되어서 매일 비현실적인 기대치에 대한 주요 진척도를 보고해야 한다는 압박을 느끼게 된다. 또한 팀은 문제점을 보고하는 것에 대해서 억제를 당한다고 느끼며 팀이 스스로 잘 관리하지 못하고 세부사항까지 관리를 받게 된다. 이해관계자들은 팀의 회의에 따라가기 보다는 팀의 진척을 느리게 하는 장애 요소들을 제거해 주는 것이 더 유익할 것이다.

스프린트 동안 진척도 추적(Tracking Progress during the Sprint)

스크럼에서 팀은 스스로 관리한다. 팀은 이를 성공적으로 하기 위해서 어떻게 스스로 관리를 하는지를 알아야 한다. 매일 팀원들은 **스프린트 백로그(Sprint Backlog)**에 있는 현재의 작업을 완료하기 위해서 작업이 얼마나 남았는지를 새로 추정한다(그림 6). 또한 누군가는 종종 팀 전체로서 남은 작업을 더해서 **스프린트 번다운 차트(Sprint Burndown Chart)**에 표시한다(그림 7, 그림 8). 이 차트는 매일 팀이 완료할 작업이 얼마나 남았는지에 대한 새 추정치를 보여준다. 이상적으로 이 차트는 스프린트의 마지막 날에 “남은 작업이 0”이 되는 방향의 감소 그래프이다. 그러므로 이것은 **소멸 차트(burndown chart)**라고 불린다. 제품 개발의 현실에서는 이 그래프가 좋게 나타나는 경우도 있지만 보통은 그렇지 않다. 중요한 것은 이 차트는 팀이 목표를 향해서 어느 정도 나아갔는지를 보여준다는 것이다. 즉, 팀이 과거에 얼마의 시간을 썼는지가 아니라 팀을 목표에서 떼어놓는 일들이 앞으로 얼마나 남았는가를 보여준다. 만약 번다운 차트의 그래프가 스프린트가 끝날 때 완료가 될 방향으로 가고 있지 않다면 팀은 계속 지속 가능한 속도를 유지하면서 일의 범위를 줄인다던가 또는 더 효율적으로

일할 방향을 찾아서 조정해야 한다.

스프레드시트를 이용해서 스프린트 번다운 차트를 만들고 보여줄 수 있지만 많은 팀들은 업무 공간의 벽에 종이를 붙이고 펜으로 적는 것이 더 효율적이라는 것을 알고 있다. 이런 “기술이 많이 필요하지 않은 고감도(low-tech/high-touch)”의 해결책이 보통 컴퓨터 차트보다 빠르고 간단하며 시각적으로 더 좋다.

제품 백로그 아이템	스프린트 작업	지원자	초기 업무량 추정치	업무의 양에 대한 새로운 추정치 하루가 끝날 때 남은 것...					
				1	2	3	4	5	6
구매자로서, 나는 쇼핑카트에 책 한 권을 넣고 싶다.	데이터베이스 수정	Sanjay	5						
	웹페이지 생성(UI)	Jing	8						
	웹페이지 생성(자바스크립트 로직)	Tracy & Sam	13						
	자동화된 인수 시험 작성	Sarah	13						
	구매자 도움 웹페이지 업데이트	Sanjay & Jing	3						
...									
트랜잭션 처리 성능 향상	DCP 코드 병합과 레이어 수준 테스트 완료		5						
	자동화된 인수 시험 작성		8						
	구매자 도움 웹페이지 업데이트		13						

그림 6. 스프린트 백로그에 남아있는 매일 새롭게 갱신되는 업무들

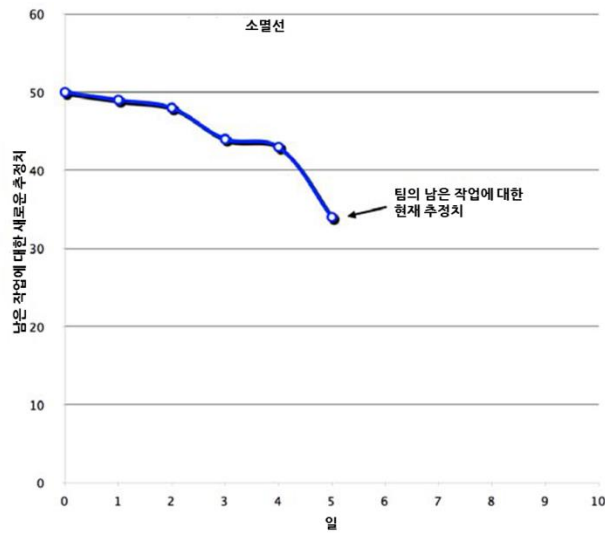


그림 7. 스프린트 번다운 차트

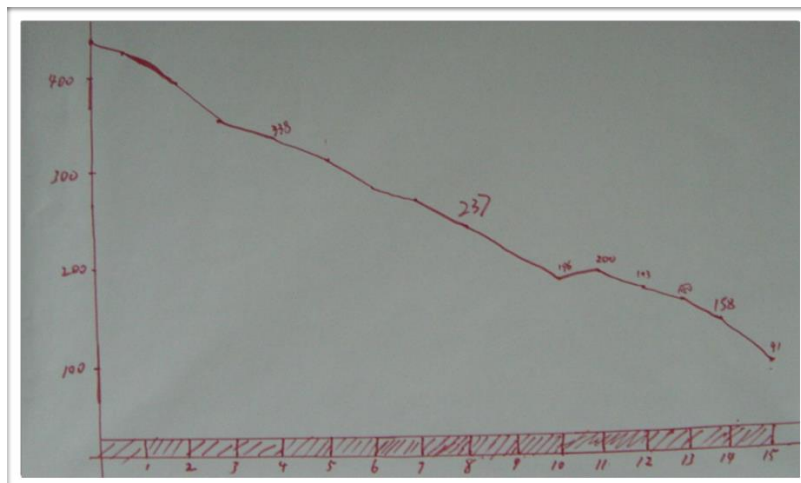


그림 8. 시각적 관리: 손으로 그린 스프린트 번다운 차트

제품 백로그 개선(Product Backlog Refinement)

요약: 미래의 스프린트를 위해서 큰 아이템을 쪼개고 아이템을 분석하고 다시 추정하고 우선순위를 다시 정함

참가자: 팀은 참가. 제품 책임자는 만약 세부적인 개선에 대해서 도움을 줄 수 있는 전문가라면 전체 활동에 참가함. 요구사항을 이해하고 팀에 도움을 줄 수 있는 사람들 참가. 스크럼마스터는 초반에 효과적으로 진행할 수 있도록 지도하기 위해서 참가하고 나머지 부분은 참가하지 않을 수도 있음.

기간: “분석이 작업이 많은” 아이템이라서 더 오래 걸릴 수 있더라도 보통 스프린트에서 팀이 사용 가능한 시간의 10%를 넘지 않도록 함.

많이 알려지진 않았지만 가치 있는 스크럼의 가이드라인 중 하나는 매 스프린트의 일정 부분은 팀 전체가 미래의 스프린트들을 위해서 제품 백로그를 개선하는데 써야 한다는 것이다. 여기에는 세부적인 요구사항 분석, 큰 아이템을 작은 아이템으로 쪼개는 것, 새로운 아이템에 대한 추정을 하는 것, 현재의 아이템들을 다시 추정하는 것을 포함된다. 스크럼에서는 어떻게 일을 해야 하는지에 대해서는 언급하지 않지만 자주 사용되는 기법 중 하나로는 스프린트의 중반이나 끝에 집중적인 워크샵을 가져서 팀과 제품 책임자 그리고 다른 이해관계자들이 방해 받지 않고 이 개선업무에 집중할 수 있도록 하는 것이 있다.

이런 개선활동은 현재 스프린트의 아이템을 위한 것이 아니다. 이 활동은 대개 한 두 스프린트 다음이 될 미래의 아이템을 위한 것이다. 이런 활동을 통해서 스프린트 계획은 상대적으로 간단해진다. 왜냐하면 제품 책임자와 스크럼 팀은 명확하고 분석이 잘 되었으며 신중하게 추정된 아이템들에 대한 계획을 하기 때문이다. 스프린트 계획에서 중요한 질문이나 발견, 혼란이 있거나 완성이 덜 된 느낌이 든다는 것은 이 개선 워크샵이 끝나지 않았거나 잘 되지 않았다는 신호이다. 이렇게 되면 계획 업무가 스프린트 자체로 번져나가게 되며, 이는 일반적으로 바람직하지 않다.

스프린트 리뷰(Sprint Review)

요약: 제품 증분의 기능성에 관한 검토와 조정

참가자: 팀, 제품 책임자, 스크럼마스터, 제품 책임자가 초대할 적합한 이해관계자들.

기간: 스프린트의 한 주당 한 시간으로 제약

스프린트가 끝나면 **스프린트 리뷰**를 하며 여기서 사람들은 스프린트를 검토한다. 이 회의의 참가자는 제품 책임자, 팀원, 스크럼마스터 그리고 추가적으로 고객, 사용자, 이해관계자, 전문가, 경영진 그리고 관심이 있는 누구나가 될 수 있다. 2주 길이의 스프린트는 최대 2시간 이내에 끝낸다. 참가하는 누구나 자유롭게 질문할 수 있고 의견을 줄 수 있다.

리뷰는 종종 “데모”라는 잘못된 꼬리표를 달기도 하지만 이는 이 회의의 진정한 의도를 모르는 것이다. 스크럼의 중요한 개념은 *검토와 조정*이다. 반복되는 주기 안에서 무엇이 진행되고 있는지 보고 배우며 피드백에 기반해서 발전시켜 나간다. 스프린트 리뷰는 *제품*을 위해서 검토하고 조정하는 활동이다. 이 때 제품 책임자는 제품과 팀에 대해서 상황이 어떤지 배운다(이것이 바로 스프린트에 대한 검토이다). 또한 이 때 팀도 제품 책임자와 시장에 대해서 어떻게 되어가고 있는지 배운다. 따라서 리뷰에서 중요한 요소는 팀과 제품 책임자간의 상황에 대해서 배우고, 조언을 구하는 등의 깊이 있는 *대화*이다.

리뷰는 분명 팀이 스프린트 동안 만든 실제로 동작하는 소프트웨어를 사용하는 것을 포함하지만 리뷰의 초점이 대화를 하는 것 보다는 단지 제품을 보는 것에만 집중된다면 균형이 맞지 않다.

스프린트 리뷰에서 “살아있는 소프트웨어”라는 것은 팀이 제시하는 “프레젠테이션”이 아니며 또한 슬라이드웨어를 쓰지 않는다. 이는 예를 들면 샌드박스 개발 환경 같은 실제로 동작하는 소프트웨어를 전달해서 검토한다는 의미이다. 리뷰를 하는 방에는 사람들이 동작하는 소프트웨어를 검토하고 사용해 볼 수 있도록 한 대 이상의 컴퓨터를 둔다. 리뷰에서는 팀이 수동적으로 데모를 하는 것 보다는 실제 사용자와 제품 책임자가 소프트웨어를 전달받아서 소통할 수 있는 활동적인 활동을 더 권장한다.

스프린트 리뷰를 준비하는데 30분을 넘지 않도록 해야 하며 그렇지 않다면 무언가 잘못된 것이다.

스프린트 회고(Sprint Retrospective)

요약: 프로세스와 환경에 관련된 것을 검토하고 조정하는 활동

참가자: 팀, 스크럼마스터, 제품 책임자(선택). 팀이 초대하지 않은 이해관계자들은 참석 불가

기간: 스프린트 한 주 당 45분으로 제한

스프린트 리뷰는 *제품*에 대해서 검토하고 조정하는 것을 포함한다. 리뷰 후에 연이어 하는 **스프린트 회고**는 *프로세스와 환경*에 대해서 검토하고 조정하는 것을 포함한다. 이것은 팀이 어떤 것이 효과가 있고 어떤 것이 효과가 없는지 토론하고 무엇을 바꿀지에 대해서 합의하기 위한 자리이다. 때로 스크럼마스터는 회고에서 퍼실리테이터(facilitator) 역할을 할 수 있다. 하지만 회고를 위해서 중립적인 외부 퍼실리테이터를 찾는 것이 더 나을 수도 있다. 스크럼마스터가 각자의 회고를 퍼실리테이팅 해주는 것은 좋은 접근방식이며 이것을 통해서 팀원들은 서로 소통할 수 있다.

스프린트 회고를 진행하는 기법은 여러 가지가 있고 *Agile Retrospectives*(Derby, Larsen 2006)이라는 책이 여러 유용한 기법들을 제공한다.

많은 팀들이 회고를 단지 문제에만 집중해서 진행하는데, 이것은 잘못되었다. 이런 방식은 사람들이 회고를 다소 우울하고 부정적인 활동이라고 생각하게 만든다. 이런 방법 대신 매 회고가 긍정적인 부분 또는 강점에 또한 집중할 수 있도록 해야 한다. *Appreciative Inquiry* 사이트에 있는 여러 책들을 통해서 더 자세한 팁들을 얻을 수 있다.

항상 같은 분석 기법을 쓰는 회고는 지루해질 수 있다. 그러므로 시간이 지나면서 다양한 기법들을 소개해야 한다.

다음 스프린트의 시작(Starting the next Sprint)

스프린트 리뷰 후 제품 책임자는 새로운 시각으로 새로운 아이템을 추가하고 더 이상 쓸모 없는 부분은 제거하며 현재 있는 것들을 수정하고 제품 백로그를 업데이트한다. 업데이트된 제품 백로그의 예시인 그림 9를 보자.

		새로운 추정치 스프린트가 끝날 때 남은 것...							
우선순위	아이템	세부사항 (위키주소)	초기 추정치	1	2	3	4	5	6
1	구매자로서, 나는 책 한 권을 쇼핑카트에 담고 싶다 (위키 페이지에 있는 UI 스케치를 볼 것)	...	5	0	0	0			
2	구매자로서, 나는 쇼핑카트에 있는 책 한 권을 제거하고 싶다	...	2	0	0	0			
3	트랜잭션 처리 성능 향상(위키에 있는 목표 성능치 측정기준을 볼 것)	...	13	13	0	0			
4	신용카드 승인 속도 향상을 위한 솔루션 조사(위키에 있는 목표 성능치 측정기준을 볼 것)	...	20	20	20	0			
5	모든 서버를 아파치 2.2.3으로 업그레이드	...	13	13	13	13			
6	스크립트 에러를 처리하는 순서 진단, 결정(bugzilla ID 14823)	...	3	3	3	3			
7	쇼핑객으로서, 나는 위시 리스트를 만들고 저장하고 싶다	...	40	40	40	40			
8	쇼핑객으로서, 나는 내 위시 리스트에 아이템을 넣거나 빼고 싶다	...	20	20	20	20			
...			537	580	570	500			

그림 9. 업데이트된 제품 백로그

스프린트 사이에는 휴식 시간이 없다. 팀들은 보통 어느 날 오후에 스프린트 회고를 하고 바로 다음날 아침에 스프린트 계획을 한다. (또는 주말 후)

애자일 개발의 한 가지 원칙은 “지속 가능한 속도”이며 탐은 합리적인 수준으로 적절히 일해야 이 주기를 계속 이어나갈 수 있다. 생산성은 팀의 실천방식이 발전하고 팀의 생산성에 방해가 되는 요소들을 제거해 가며 계속해서 오르는 것이지, 잔업이나 품질에 대해서 타협하며 오르는 것이 아니다.

스프린트는 제품 책임자가 제품을 출시할 준비가 되었다고 판단할 때까지 계속된다. 완벽한 스크럼에서 제품은 매 스프린트가 끝날 때 마다 잠재적으로 출시가 가능한 상태이다. 이는 테스트나 문서화 같은 마무리 작업이 필요하지 않다는 뜻이다. 이 말이 함축하고 있는 것은 매 스프린트마다 모든 것이 완벽하게 끝나야 한다는 것이다. 즉, 리뷰가 끝난 후 즉시 실제로 출시 가능하거나 배포할 수 있어야 한다. 그러나 많은 조직들은 개발 방법, 툴, 인프라가 약하고 이런 완벽한 스크럼은 할 수 없으며 그렇기 때문에 남은 일을 처리하기 위해서 “릴리즈 스프린트(Release Sprint)”가 필요하다. “릴리즈 스프린트”가 필요할 때, 이것은 필요악이며 조직은 실천방식의 개선을 통해서 더 이상 릴리즈 스프린트가 필요 없도록 해야 한다.

릴리즈 관리(Managing Release)

때로 오는 질문 중 하나는 반복적인 모델의 경우 장기적인 출시 계획을 어떻게 해야 하나냐는 것이다. 생각해 봐야 할 것이 두 가지가 있다. (1) 첫 번째 릴리즈를 위한 새로운 제품 (2) 추후의 릴리즈를 위한 현재의 제품.

새로운 제품의 경우나 *현재의 제품에 단지 스크럼만 적용*하는 경우, 첫 번째 스프린트 전에 제품 백로그 개선을 시작할 필요가 있으며, 여기서 제품 책임자와 팀은 적절한 스크럼 제품 백로그를 만든다. 이 작업은 며칠이 걸릴 수도 있고 한 주가 걸릴 수도 있으며 워크샵을 하게 될 수도 있다. 이 워크샵은 때론 초기 제품 백로그 제작(Initial Product Backlog Creation) 또는 릴리즈 계획(Release Planning)라고 불린다. 그리고 세부적인 요구사항 분석이나 첫 번째 릴리즈 때 확인할 모든 아이템에 대한 추정도 필요하다.

놀랍게도 스크럼에서 확실히 정해진 제품 백로그가 있는 제품의 경우는 다음 릴리즈를 위한 어떤 특별하거나 큰 릴리즈 계획이 필요 없다. 왜 그럴까? 왜냐하면 제품 책임자와 팀은 매 스프린트마다 5 ~ 10% 정도의 시간을 제품 백로그 개선에 써야 하고 계속해서 추후를 위해 준비해야 하기 때문이다. 이런 지속적인 제품 개발 방식에서는 기존의 순차적인 수명 주기를 따르는 개발에서 볼 수 있었던 준비-실행-종료 단계의 극단적으로 끊어지는 단계에 대한 준비가 필요 없다.

최초의 제품 백로그 개선 워크샵과 매 스프린트의 백로그 개선을 하는 동안, 팀과 제품 책임자는 릴리즈 계획도 세우고 학습한 것을 바탕으로 추정치와 우선순위도 개선도 할 것이다.

어떤 릴리즈는 데이터가 중심이 된다. 예를 들면 “우리는 프로젝트의 2.0 버전을 11월 10일에 무역 박람회에서 출시할 것이다.”가 될 수 있다. 이런 상황의 경우, 팀은 시간이 되는대로 최대한 많은 스프린트를 완료하고 기능(feature)을 만들 것이다. 다른 제품들은 완료라고 할 수 있기 전에 특정한 기능들을 요구하며 제품들은 이런 요구사항들이 얼마나 시간이 걸리든 만족시키기 전에는 출시가 될 수 없을 것이다. 스크럼은 매 스프린트마다 잠재적으로 출시 가능한 코드를 만드는 것을 강조하기 때문에 제품 책임자는 중간 릴리즈를 하기로 할 수도 있고, 고객이 완료된 작업에 대한 수혜를 더 빨리 받을 수 있게 할 수도 있다.

사람들은 다가오는 모든 일들을 알 수 없기 때문에, 계획을 세우고 개선하는 것을 통해서 릴리즈에 대한 큰 방향성을 제시하고 어떻게 결정들의 균형을 맞추는지(예를 들면 범위나 일정이나 같은)를 명확히 하는 것이 핵심이다. 이것을 당신을 최종 목적지로 안내해주는 로드맵이라고 생각해라. 당신이 정확히 어떤 길을 가고 어떤 결정을 내릴지는 여행을 하는 도중에 결정될 것이다.

여행보다는 목적지가 더 중요하다.

대부분의 제품 책임자들은 한 번의 릴리즈만 하는 방식을 선택한다. 예를 들면, 제품 책임자는 릴리즈 날짜를 결정할 것이고 그 날짜까지 완료될 수 있는 제품 백로그 아이템을 추정하기 위해서 팀과 함께 일할 것이다. 계약이 있는 개발과 같이 “고정된 가격 / 고정된 날짜 / 고정된 제품”과 같은 투입이 있는 경우는 예상치 못한 것이나 변화에 대응할 수 있도록 여유를 뒀야 한다. 이런 점에서 스크럼은 다른 접근방식과 다르지 않다.

애플리케이션 또는 제품에서의 관점

애플리케이션이나 제품이 시판하기 위한 것이든 내부적으로 사용하기 위한 것이든, 스크럼은 *지속적인 애플리케이션/제품 개발* 모델을 위한 오래된 *프로젝트* 중심의 모델에서 탈피한다. 더 이상 시작, 중간, 끝이 있는 프로젝트는 없다. 그리고 이런 이유로 기존의 프로젝트 관리자도 없다. 오히려 단순하게 고정된 제품 책임자와 제품이나 애플리케이션 개발이 끝날 때까지 길이가 고정된 스프린트 동안 “끝없이” 협업하며 오래 유지되는 자기관리 팀이 존재한다. 필요한 모든 “프로젝트” 관리 업무는 팀과 제품 책임자가 한다. 여기서의 제품 책임자는 내부적인 사업 고객이거나 제품 관리와 관련된 사람이다. 이런 관리 업무는 IT 관리자나 프로젝트 관리 쪽의 사람이 하지 않는다.

스크럼은 또한 오랜 기간 동안 사용될 애플리케이션을 만들고 발전시키기 위한 업무가 아닌, 단발적인 계획을 가진 진정한 의미에서의 *프로젝트*에도 활용될 수 있다. 이런 경우에도 여전히 팀과 제품

책임자가 프로젝트를 관리한다.

만약 현재 존재하는 애플리케이션들 각각에 대해서 새로운 업무가 충분하지 않아 각 애플리케이션에 전념하는 팀들이 오래 유지된다는 것을 보장할 수 없다면 어떻게 할 것인가? 이런 경우 고정적으로 오래 유지되는 팀은 한 스프린트마다 한 개의 애플리케이션에 대한 아이тем들을 맡고 다른 애플리케이션에 대한 아이тем은 다음 스프린트에 할 수 있다. 이런 상황이라면 스프린트는 보통 한 주 같은 짧은 주기를 가진다.

때로는 앞의 방법으로도 해결하기 힘들 정도로 새로운 일이 적어서 팀이 한 스프린트 내에 여러 애플리케이션의 아이тем들을 담당할 수도 있다. 그러나 이런 방법은 다양한 애플리케이션에 걸쳐서 비생산적인 멀티태스킹을 만들 수도 있으니 주의해야 한다. 스크럼에서 생산성에 대한 기본적인 한 가지는 팀은 한 스프린트에 한 개의 제품이나 애플리케이션에 집중해야 한다는 것이다.

공통 과제(Common Challenges)

스크럼은 그저 구체적인 실천 방안들을 묶어 놓은 것이 아니다. 스크럼은 투명성을 제공해주는 프레임워크이며 “검토와 조정”을 가능하게 해주는 메커니즘이라는 것이 중요하다. 스크럼은 제품 책임자와 팀의 효율에 영향을 주는 장애 요소들을 시각적으로 나타나게 해서 그 부분들을 고민할 수 있도록 해준다. 예를 들면 제품 책임자는 실제로 시장에 대해서, 기능(feature)에 대해서, 사업적 가치와 관련된 것들을 어떻게 측정하는지에 대해서는 모를 수도 있다. 또는 팀이 일을 추정하는 것이나 개발을 하는 것에 서툴 수도 있다.

스크럼 프레임워크는 빠르게 이런 약점들이 드러나도록 할 것이다. 스크럼은 개발에서의 문제를 해결하지는 못한다. 하지만 이런 문제점들이 괴로울 정도로 잘 드러나게 해주고 사람들이 짧은 주기 내에 개선을 위한 작은 실험들을 하며 문제를 해결할 방법을 찾을 수 있도록 프레임워크를 제공한다.

팀이 업무 분석과 추정 기술이 미흡해서 첫 번째 스프린트에 예측한 것을 완료하지 못했다고 가정해보자. 그러면 팀은 실패자가 된 느낌을 받게 된다. 그러나 실제로 이런 경험은 더 현실적이고 깊이가 있는 예측을 하기 위해서 필요한 첫 걸음이다. 장애 요소를 잘 보이게 하고 팀이 그것에 대처할 수 있게 도와주는 이런 스크럼의 패턴은 팀이 스크럼을 적용하면서 얻을 수 있는 가장 큰 이점을 만들어주는 메커니즘이다.

도전적인 스크럼 실천 방법을 할 때의 한 가지 흔한 실수는 스크럼을 변형시키는 것이다. 예를 들어 산출물을 만드는데 문제가 있는 팀은 스프린트의 기간을 늘일 수도 있고 그래서 절대 시간이 다 되지 않을 것이다. 그리고 그런 프로세스에서는 추정하고 시간을 관리하는 것을 어떻게 하면 더 잘할지 절대 배울 필요가 없을 것이다. 이런 식으로 경험이 있는 스크럼마스터의 코칭이나 지원 없이 조직은 그들의 약점과 장애와는 반대로 스크럼을 변형시킬 수도 있고 스크럼이 주는 진정한 혜택을 약하게 만들 수도 있다. 여기서 말하는 스크럼이 주는 진정한 혜택은 장점과 단점을 눈에 보이도록 하고 조직이 더 높은 수준으로 나아가기 위한 선택 사항을 제시하는 것을 뜻한다.

다른 흔한 실수는 스크럼이 특정 실천 사항을 분명하게 요구하지 않는다는 이유로 어떤 실천 사항을 피하거나 금지해야 한다고 생각하는 것이다. 예를 들면, 스크럼에서는 제품 책임자가 제품을 위한

장기적인 전략을 세우는 것을 요구하지 않는다. 또한 스크럼에서는 엔지니어가 복잡한 기술적인 문제에 대해서 더 경험이 많은 엔지니어에게 조언을 구하는 것을 요구하지도 않는다. 스크럼에서는 이런 부분들은 개인이 올바른 결정을 하도록 남겨둔다. 대부분의 경우는 위의 두 실천 사항은 하는 것이 더 낫다.

주의해야 할 것은 관리자가 팀에 스크럼을 도입하는 것이다. 스크럼은 팀에게 스스로 관리하기 위한 공간과 도구를 제공한다. 위에서부터 이런 것들이 좌우되는 것은 성공을 위한 길이 아니다. 더 나은 접근 방법은 팀이 동료나 관리자에게 스크럼에 대해서 배우고, 전문적인 교육을 통해서 포괄적으로 배우고, 그리고 나서 팀 차원에서 정해진 기간에 실천 사항들을 제대로 따를 것을 결정한다. 기간이 끝날 즈음, 팀은 경험한 것을 평가해보고 계속 할 것인지 결정한다.

좋은 소식은 첫 번째 스프린트는 보통 팀에게 매우 도전적이지만 이 스프린트가 끝날 때쯤에는 스크럼의 이점이 보이는 편이며, 많은 스크럼 팀들이 “스크럼은 어렵지만 이전에 우리가 하던 것 보다는 훨씬 더 좋아졌어!” 라고 외치도록 이끌어준다는 것이다.

부록 A: 추가적인 자료

스크럼에 대해서 출판된 자료는 많다. 여기서 우리는 몇 가지 추가적인 온라인 자료나 책들을 알려주고자 한다.

온라인 자료:

- [The Lean Primer – An introduction to Lean Thinking, an important influence to Scrum.](http://www.leanprimer.com)
<http://www.leanprimer.com>
- [The Distributed Scrum Primer – Additional tips for teams who aren't co-located.](http://www.goodagile.com/distributedscrumprimer/)
<http://www.goodagile.com/distributedscrumprimer/>
- [The ScrumMaster Checklist – A list of question that good ScrumMasters use.](http://www.scrummasterchecklist.org/)
<http://www.scrummasterchecklist.org/>
- [Feature Team Primer – Scaling Scrum with Feature Teams.](http://www.featureteams.org)
<http://www.featureteams.org>
- [The Agile Atlas – Core Scrum. ScrumAlliance description of Scrum.](http://agileatlas.org/atlas/scrum)
<http://agileatlas.org/atlas/scrum>
- [Scrum Guide – Scrum.org description of Scrum.](http://www.scrum.org/Scrum-Guides)
<http://www.scrum.org/Scrum-Guides>
- [Agile Contracts Primer – How to make Scrum-friendly contracts.](http://www.agilecontracts.org/)
<http://www.agilecontracts.org/>

서적:

- Leading Teams – Richard Hackman
- Scaling Lean & Agile Development: Thinking and Organizational Tools for Large-Scale Scrum – Craig Larman, Bas Vodde
- Practices for Scaling Lean & Agile Development: Large, Multisite, and Offshore Product Development with Large-Scale Scrum – Craig Larman, Bas Vodde
- Agile Project Management with Scrum – Ken Schwaber
- Succeeding with Agile: Software Development using Scrum – Mike Cohn

부록 B: 용어

번다운(Burn Down)

스프린트, 릴리즈, 제품의 기간별 남은 작업량에 대한 추세. 가공하지 않은 데이터는 스프린트 백로그와 제품 백로그에서 나오며 남은 일은 세로축에, 기간(하나 또는 여러 스프린트의 일수)은 가로축에 나타난다.

데일리 스크럼(Daily Scrum)

팀에 의해서 매일 열리는 짧은 회의. 여기서 팀원들은 작업에 대해서 검토를 하고 서로 작업과 진척도를 동기화하며 장애 요소에 대해서 제거할 수 있도록 스크럼마스터에게 말한다. 스프린트를 최적화하기 위해서 필요한 곧 해야 할 작업들을 조정하기 위한 회의는 데일리 스크럼 후 열릴 수 있다.

개발팀(Development Team)

역할을 위한 팀의 다른 이름

완료(Done)

모든 참여자가 서로 동의하고 조직의 기준, 관습, 가이드라인에 맞는 완료. 스프린트 리뷰 회의에서 “완료”라고 무언가 보고되었을 때, 이것은 반드시 합의된 정의에 부합해야 한다.

추정된 남은 작업(스프린트 백로그 아이템)

팀원들이 추정한 어떤 업무를 하기 위해서 남은 시간. 이 추정은 스프린트 백로그의 업무를 할 때 매일 하루가 끝날 때 업데이트된다. 여기서 추정치는 그 업무를 몇 명이 하는가에 관계없이 남은 전체 업무의 양을 추정한 것이다.

증분(Increment)

팀이 매 스프린트마다 개발한 잠재적으로 출시가 가능하거나 제품 책임자의 이해관계자들에게 쓸모가 있는 제품의 기능.

잠재적으로 출시 가능한 제품의 기능에 대한 증분(Increment of Potentially Shippable Product Functionality)

제품 전반이나 시스템의 완전한 조각 중 제품책임자 또는 이해관계자가 구현하기로 선택했을 경우 사용할 수 있게 될 완전한 일부분.

스프린트(Sprint)

제품 또는 시스템의 증분을 만드는 반복 또는 유사한 업무를 하는 반복되는 하나의 주기. 한 달이 넘지 않으며 보통 한 주는 넘는다. 전체 작업 동안 기간은 고정되어 있으며 같은 시스템 또는 제품에 대한 팀들은 같은 길이의 주기를 가진다.

제품 백로그(Product Backlog)

완료된 제품 기능이 되기 위한 추정치를 가진 우선순위화된 요구사항 리스트. 아이템이 제품 백로그에서 우선순위가 더 높을수록 추정치가 더 정확하다. 이 리스트는 사업 상황이나 기술 변화에 따라서 변하며 도출된다.

제품 백로그 아이템(Product Backlog Item)

기능적 요구사항, 비기능적 요구사항, 이슈, 사업 중요성에 따라 우선순위화된 것들, 종속된 것들, 추정된 것들. 추정에 있어서의 정밀함은 제품 백로그 아이템의 우선순위와 세분화에 달려있다. 다음 스프린트에 선택 될 가능성이 있는 높은 우선순위의 아이템은 매우 세분화되어야 하고 정밀해야 한다.

제품 책임자(Product Owner)

제품의 가치를 최대화하기 위해서 제품 백로그에 대한 책임이 있는 사람. 제품 책임자는 프로젝트에서 이해관계가 있는 모든 사람의 이익을 대변하는 것과 결과물인 제품에 대한 책임이 있다.

스크럼(Scrum)

약어가 아니라 럭비 경기에서 장외로 나간 공을 다시 경기장 안으로 들이기 위한 메커니즘.

스크럼 마스터(ScrumMaster)

스크럼 프로세스와 그것을 알맞게 이행하고 이점을 최대화 하는데 책임이 있는 사람.

스프린트 백로그(Sprint Backlog)

스프린트에서 팀을 위한 업무의 리스트. 이 리스트는 종종 더 세분화된 업무들로 분해된다. 스프린트 계획을 하는 동안 리스트를 도출하고 팀이 스프린트를 진행하는 동안 필요에 따라 아이템들을 제거하거나 새로운 업무를 추가하면서 업데이트 될 수도 있다.

스프린트 백로그 작업(Sprint Backlog Task)

팀이나 팀원이 전담하는 제품 백로그 아이템을 시스템 기능으로 바꾸기 위해서 필요하다고 정의한 작업 중 하나.

스프린트 계획 회의(Sprint Planning meeting)

매 스프린트를 시작하는 4시간 이내(2주 단위의 스프린트 기준) 의 회의. 이 회의는 두 시간 단위의 두 개의 회의로 나뉘며 각각의 회의 또한 시간이 제한된다. 첫 번째 파트 동안 제품 책임자는 우선순위가 가장 높은 제품 백로그를 팀에게 준다. 팀과 제품 책임자는 협업을 통해서 팀이 다가오는 스프린트 동안 얼마만큼의 제품 백로그를 기능으로 바꿀지 결정하는 것을 돕는다. 두 번째 파트에서 팀은 앞에서 결정한 것을 설계와 분해를 통해서 어떻게 성공시킬지를 계획한다. 이를 통해서 팀은 어떻게 스프린트 목표를 달성할 것인지를 이해하게 된다.

스프린트 회고 회의(Sprint Retrospective meeting)

스크럼마스터의 퍼실리테이션을 통해서 완전한 팀이 막 끝난 스프린트에 대해서 토론하고 다음 스프린트를 더 즐겁고 생산적으로 만들기 위해서 무엇을 바꿀지를 결정하는 회의.

스프린트 리뷰 회의(Sprint Review meeting)

매 스프린트가 끝날 때 하는 두 시간 이내(2주 단위의 스프린트 기준)의 회의로, 여기서 팀은 제품 책임자와 이해관계자들과 함께 회의를 하고 이들은 스프린트의 산출물을 검토한다. 이 회의는 보통 완료된 제품 백로그 아이템에 대한 리뷰, 기회에 대한 토론, 제약 사항, 리스크 또는 다음 번에 무엇이 가장 중요할지에 대한 토론(잠재적으로 제품 백로그의 변화를 가져옴)으로 시작된다. 오직 완성된 제품 기능만 시연할 수 있다.

이해관계자(Stakeholder)

투자를 했거나 프로젝트의 결과물을 사용할 것이거나 또는 프로젝트의 결과물에 영향을 받기 때문에 프로젝트의 결과에 관심이 있는 사람.

팀(Team)

매 스프린트마다 제품의 증분을 개발하기 위해서 스스로를 관리할 책임이 있는 다기능적인 무리의 사람들.

시간 제한(Time box)

활동이나 회의를 할 때 넘어서는 안 되는 시간의 길이. 예를 들면 데일리 스크럼 회의는 15분의 시간 제한이 있으며 15분이 끝나면 어찌 되었건 마친다. 회의의 경우는 이 시간보다 짧게 해야 하며 스프린트의 경우는 정확하게 그 시간만큼 한다.