

The Impact of Agile/Scrum on Requirements Management

요구사항 관리 관점에서 애자일/스크럼의 영향력

00:00 – 07:15

이 프리젠테이션은 작년에 코펜하겐에서 발표되었던 내용을 요구사항 관리를 위해 조금 편집하였습니다. 강연이 유익하셨으면 좋겠습니다. 저는 현재 요구사항 실천 커뮤니티 매니저(requirement practice community manager)로 캐나다의 브리티시 컬럼비아에 본사가 있는 SERA 시스템즈라는 컨설팅 회사에서 일하고 있습니다. 그래서 저는 많은 원격회의와 아름다운 밴쿠버로의 여행을 많이 가곤 합니다. 하지만 저는 대부분 다른 해안에서 오기 때문에 대부분 바다만을 보고 있습니다. 여기까지가 제 소개였구요, 이 프리젠테이션이 원래 주제에서는 조금 벗어나서 새로운 주제로 발표되는 것을 양지하시고, 들으셨으면 좋겠습니다. 이제 시작하겠습니다.

이미 말씀 드린 대로 요구사항 관리의 관점에 맞추기 위해 원래의 프리젠테이션에서 조금 수정되었습니다. 원래는 커뮤니티 의회에서 일하는 제 동료 Beth Ouellette 에 의해 코펜하겐에서 발표된 자료입니다. 프리젠테이션의 주제는 Agile Scrum 프레임워크가 어떻게 특정 시스템과 함께 요구사항 관리에 있어서 프로젝트의 접근을 변화시키는가 이고, 또, 비즈니스 분석에 있어서 다양한 역할들의 효과와 사업분석 역할에 대한 요구사항 관리에 대해 다루었습니다. 의제로서, 변경에 대해 다루고, 더 나아가 배경 지식으로서 Agile 에 대해 알아볼 것입니다. 또, Scrum 의 개념과 역할에 대해 얘기하게 될 텐데, 제가 강조하고 싶은 건 스크럼이고, 이미 많은 분들은 스크럼을 사용한 경험이 있다는 걸 알고 있습니다. 아마 50% 정도일 것 같고, 요구사항 관리에 주는 영향에 대해 알아보고 마칠 것입니다.

먼저, 변화에 대해 좀 얘기하도록 하죠. 변화, 이게 뭘까요? 뭔가가 이루어지거나, 무엇인가를 다르게 만드는 것, 이것이 변화에 대한 모든 것일 것입니다. 확신하건대 우리 중의 대부분은 변화에 대해 경험하고 있습니다. 그리고 우리 모두가 동의할 것으로 생각되는데, 대부분의 사람들은 변화를 싫어하고 변화보다는 그대로 있는 것을 선호합니다. 또한 우리 대부분은 Agile 은 변경의 일종이라는 것에 동의할 것이고, 또한 그것은 요구사항과 다른 프로젝트의 수많은 것들과 연관되어 있다는 사실을 아실 것입니다. 그래서, 무엇인가를 대체하거나 교체할 때, 때로는 조직의 처리 방법에 관해서나 성공한 프로젝트에서 사용한 전통적인 방법들을 바꾸고자 할 때, Agile 한 방법은 변화를 고려하게 되는 것이 될 것입니다.

변경에 대한 많은 정의가 있습니다. (정리해줘서 고마워요, Beth. 프레젠테이션 중간중간에 분홍색과 파란색 이미지들이 있는데, 제가 고른 것은 아닙니다만, 어쨌든 좋아하게 되었습니다). 변경은 고문과 같습니다. 우리 모두 알고 있고, 우리의 친구 마키아벨리는 이렇게 말했습니다. "실제로 행하는 것보다 어려운 것은 없고, 행하는 것보다 위험한 것은 없고, 성공하는 것보다 더 불확실한 것은 없다. 그리고 새로운 질서를 도입하는 것에 있어서 이를 주도하는 것보다 더 한 것은 없다." 여러분 중의 많은 분들이 Agile 프로젝트를 수행하고 계시는데, 변경은 우리가 관리해야 하는 것 중의 하나입니다.

변경에 대한 대부분의 주요 영역은 긍정적인 것에 초점이 맞추어져 있고, 이는 변경에 영향을 받는 인력들, 변경을 처리하기 위한 계획, 사람들과 함께 이를 진행하고 변화를 위한 계획을 만드는 것을 포함하고 있습니다. 이런 것들이 변경에 대한 기본적인 변경관리입니다. 몇 분은 Agile 방법론에 대해 잘 아실 텐데, 저는 개인적으로 ADKRA 라는 단어를 좋아합니다. 왜냐하면 이것은 Awareness, Desire, Knowledge, Ability 그리고 Re-enforcement 의 약자이기 때문이고, 이 한 단어로 여러분들은 변경을 어떻게 해야 하는 지에 대해 알려줄 수 있습니다.

Awareness, 인지는 고위층으로부터 프리젠테이션의 형태로 오고 있습니다.. Desire, 욕구는 당신이 보고하는 대상에게서 오게 됩니다. 우리는 당신이 변화하기를 원하며, 만약 당신이 변화하지 않으면 부정적인 결과가 당신에게 일어날 수 있습니다. Knowledge, 지식은 사람들이 변경에 대처할 수 있도록 하는 훈련입니다. Ability, 역량은 변화가 일어나고 효과를 줄 수 있다는 것을 확신케 합니다.

Re-enforcement, 재강화는 인력들이 옛날 방식으로 미끄러지지 않게 해줍니다. Agile 이라는 개념은 인력들이 이해하고 있고, 교육훈련을 받는 것도 포함합니다. 하지만 여전히 여러분들은 프로젝트 인력들이 옛날 방식을 버리지 못하는 것을 보게 될 겁니다. 그래서 변경이 일어났을 때 이것을 관리하는 것이 정말로 중요합니다. 우리는 요구사항관리에 어떤 종류의 영향이 있는 지 살펴볼 것이고, 아주 작은 것들에 대해서도 요구사항을 관리하는 것도 필요합니다. 또 Agile 과 Scrum 으로 변화하는 국제적 변화도 알아야 합니다.

07:16 – 18:29

Agile 의 배경에 대해 좀 더 얘기해봅시다. 저는 프로젝트를 시작할 때 항상 이 다이어그램을 좋아하는데, 이유는 우리가 프로젝트를 시작하고 끝내는 계획을 만들고 이 경로를 따라가게 되는데, 대부분이 이런 형태를 가집니다. 가끔은 프로젝트에 결함이 생길 경우에 프로젝트의 끝에 가서 다시 시작하는 경우가 있습니다. 이런 경우 우리는 어떤 문제인 지, 현재 프로젝트가 어떤 위치에 있는지를 파악하려고 노력하는데, 프로젝트에 마지막에 있다면 이런 다이어그램처럼 실행되지 않겠쥬. 어떤 분들은 보셨겠지만, 이것은 Agile 소프트웨어 개발에 대한 선언서와도 같으며, 아직 안보신 분이라면, URL 은 아래와 같습니다. 그리고 그 것은 기본적으로 Agile 에 대해서 이야기하고 있고, 여기에 어떤 가치들이 스며들어 있는지를 보여줍니다.

사람들이 Agile 프로세스를 적용하려고 한다고 하지만, 그들은 프로세스를 사용하면서도 프로세스나 도구를 사용하지 않으려 하고, 종합적인 문서도 사용하지 않는 것을 보게 됩니다. 많은 의미를 갖는 말들이 있지만, 중요하게 강조하고 싶은 것은 왼쪽에 있는 것입니다. 개인들간의 상호작용. 프로젝트는 개인들로 이루어지게 되기 때문에 개인들간의 상호작용이 없다면 그건 프로젝트가 아니쥬. 우리가 개인들간의 상호작용을 위한 프로세스와 도구에 집중한다면 모든 의사소통 구조가 세분화될 것이고, 사람들은 단계별로 프로세스를 따르고 집중할 수 있게 됩니다. 소프트웨어는 종합적인 문서작업 위에서 기능해야 합니다.

나는 여러분들이 많은 것들을 문서로 작성하는 프로젝트에서 일하고 있다고 확신하며, Agile 은 우리가 만들고 있는 소프트웨어라고 할 수 있습니다. 우리는 왜 문서를 만들기 보다 더 직접적으로

업무를 수행하는 소프트웨어를 만드는 것에 집중하지 않을까요? 오늘날 문서작성은 필요하지만 Agile 개념에서는 필요할 때만 문서가 의미가 있습니다. 계약협상에서 고객과의 협업을 보면, 여러 명의 변호사가 이런저런 일들을 이리저리 돌리고, 여러분들을 법적인 여러 절차에 끌어들이지만 결국은 가격에 대한 싸움입니다. Agile 이 말하는 것은 고객과의 협업입니다. 우리는 지금 계약이 필요하고, 앞으로 계약을 하게 되겠지만, 강조하는 것은 고객과의 협업과 계획에 따른 변경에 대응하는 것입니다.

물론 우리는 의심할 바 없이 계획이 필요하죠. 하지만 여러분 중의 몇몇은 계획은 쓸모 없다고 인지하고 있을 수 있습니다. 계획을 세우는 것은 굉장히 중요합니다. 계획을 세우고 그것에 매달리듯이 따라야 할 필요는 없지만 Agile 에서 얘기하는 것은 변경이 일어나고, 이것을 잘 관리하려면 우리는 변경을 환영하거나 저항하려 한다는 것입니다. 기본적으로 우리는 변화를 반겨야 한다고 Agile 은 말합니다. 그리고 필요할 경우 프로젝트의 일부를 수정해야 합니다. 그래서 Agile 은 사업적 관점에 집중하고, 소프트웨어를 개발하여 납품하고, 사람들 사이의 상호작용에 집중케 하는 하나의 관념입니다. 그리고 우리는 조금 더 빠르게 소프트웨어를 갖게 되겠죠. 그리고 업무의 우선순위를 부여하고, 적용하는 데 대한 훌륭한 사례들이 있습니다.

산에 위치한 리조트에 종사하는 Jim High Smith 는 개념들을 만든 사람들 중의 한 사람으로 이런 말을 했습니다. Agility 는 격변하는 비즈니스 환경에서 수익을 내기 위해 변화를 만들고 대응하는 능력입니다. Agile 은 PMST chapter 의 top-down 모델에서 각 팀의 리더십에 대한 역할로 진화하는데, 이것의 목적은 수행에 방해가 되는 장벽을 제거하고, 업무에 방해가 되거나 업무에 집중하지 못하게 하는 상황을 해결하기 위한 것들입니다. 프로젝트의 성공을 위해 비전을 공유하기 위한 의사소통환경을 만들기 위해서는 top-down 방식의 리더십은 Agile 모델이 아닙니다. Agile 은 상위 개념의 하나인데, 이제 그 컨셉에 대해서 보기로 하죠.

Scrum 이라는 것은 상호작용하는 개발, 지속적인 피드백 순환, 이슈에 대응하는 것들이라고 할 수 있고, 또는 Stand-up 미팅이라고 할 수도 있습니다. 제 친구 Raymond 는 스시 접근법이라고도 하고 자기관리 팀이라고 할 수도 있습니다. 프로젝트 팀은 그들이 개발하여 납품해야 하는 것이 무엇이고, 고객이 필요로 하는 것이 무엇인지, 비즈니스에서 필요로 하는 것이 무엇인지 기본적으로 주시하고 있습니다. 그리고 자기 자신들을 관리하고 있는데, 이게 의미하는 것은 PM 이 그들 위에서 군림하면서 이걸 해라 저걸 해라 하는 것이 아니라는 겁니다. 그들은 무엇을 해야 하는 지 알고 있습니다. 이것에 대해 조금 더 알아보죠.

이것이 Scrum 방법론인데, 몇 분은 아마도 좀 친숙하게 느낄 수도 있겠습니다. 우리는 제품을 만드는 시간표가 있습니다. 이것은 보통 스토리라고 불리는 우선순위를 매기는 리스트입니다. 그리고 고객들은 그 리스트를 원합니다. back log 는 이 시간표에서 뽑아낸 목록으로, 이는 Sprint 로 불립니다. 빠르게 달려야 한다는 거죠. 큰 녹색 원은 이 Sprint 를 형상화한 것인데, 앞으로 매일 2~4 주동안 Scrum 미팅을 해야 한다는 것을 알게 됩니다. 일일 Scrum 미팅이라고도 하고, Stand up 미팅이라고도 합니다.

중요한 건, 매일 같은 장소와 같은 시간에 전체 팀이 함께 모여 회의를 해야 한다는 거죠. 3 가지 만을 얘기합니다. 어제 한 일, 오늘 할 일, 그리고 나에게 오늘 어떤 일이 생길 것인지. 어떤 것도 제쳐놓고

Scrum 에서 미팅을 해야 합니다. Sprint 가 끝나면, 4 주가 지나면, 제품은 잠재적으로 납품가능한 제품으로 불립니다. 이것의 의미는 QA 도 끝났고, 테스트도 끝났고, 박스에서 보이듯이 납품이 가능하다는 거죠. 이것이 scrum 방법론의 개념입니다.

이제 여기에 있는 몇 가지 역할에 대해 알아보시다. 제품의 소유자가 있고, Scrum 관리자와 프로젝트 팀이 있습니다. 우리는 몇 가지 인위적인 결과들이 있고, 제품 backlog 가 있고, 그리고 스토리를 인지하고 있는 여러분과 고객이 있습니다. 이런 것들이 제품 backlog 이고, 여러분이 제품 backlog 에서 찾고자 하는 스토리가 있습니다. 이런 것들이 인위적인 결과물이라고 합니다.

여기서 우리가 갖고 있는 것은 프로젝트가 망가지고 있다는 것입니다. 프로젝트가 망가지고 있는 것은 현재 여기서 벌어지고 있고, 이것은 우리가 알고 있는 스토리가 망가지고 있다는 것이죠. 이게 실제 프로젝트라고 해봅시다. Scrum 안에 우리는 행사라고 부르는 것과 backlog 아이템들 그리고 카트 벽이 있습니다. 시작단계에 우리는 backlog 중에서 원하는 것을 선택하여 이 것들을 우리가 Sprint 에서 다루고자 하는 것이라고 이야기 합니다.

저는 Sprint 안에 2 주라고 언급하지는 않았는데, 여러분들이 2 주라고 선택하면 여러분은 프로젝트에 주요한 사건이 벌어지지 않는 한 Sprint 를 위해 2 주를 갖는 것입니다. 여러분이 결정하는 겁니다. 이제 Sprint 계획단계에 참여하고, 이는 여러분이 sprint 에 참가하여 다루게 될 제품의 Backlog 를 선정하고, sprint backlog 를 다시 만드는 시기입니다.

업무게시판은 기본적으로 현재에 대한 것입니다. 기본적으로 벽에 붙은 포스트잇을 한번 생각해볼 수 있으며, 그것들을 배열하면 이게 업무게시판이 되는 것입니다. Show cases 는 Sprint 의 끝에서 여러분이 한 것을 보여주는 것이며, 고객이 어떤 사람이든 초대해서 보여주더라도 당신이 Sprint 에서 한 것을 보여주는 것입니다. Sprint 회고라는 것은, Sprint 의 끝에서 일들이 마무리되고 프로젝트 팀 자체 만을 위한 것인데, 그런데 이 프로젝트 팀에는 고객과 제품 소유자와 여러분이 포함되는데, 모두가 무엇이 앞에 있고, 무엇이 일어났는 지, 잘 진행되었는 지 지켜보아야 합니다. 그리고 우리는 앞을 보고 우리가 무엇을 다시 해야 하는 지 알아야 합니다. 무엇이 변경되어야 하는지에 대해 이야기하는 것, 그것이 Sprint 회고라고 할 수 있습니다.

18:30 – 27:27

그 다음으로는 일일 Scrum 인데, 우리가 찾을 수 있는 가장 좋은 그림입니다. 좀 우습게 생각되어서 좀 크게 하였습니다. 이런 경기에 참여해본 경험이 있으신가요? 여기에 제품 Backlog 가 있습니다. 우리가 알아볼 수 있게 키웠습니다. 제가 손님으로 예약한 것을 취소했습니다. 이건 우리가 나중에 얘기할 스토리 중에 실제에 가까운 예가 됩니다. 우리가 예측에 대해 얘기해 볼텐데, 3 일이나 3 주의 일에 대한 것이 아닌 다른 것에 대한 것입니다. 이 경우에 우리는 예측하고 바로 다루어야 합니다. 이것이 제품 Backlog 가 어떻게 생겼는 지 보여줍니다. 여기에 기능도가 있구요, 여기 mingle(혼합)이라는 상품이 있습니다.

이 것은 실제 프로젝트이고, 우리는 이것들에서 기능 그리고 스토리를 보게 됩니다. 사실 이 기능들을 확장하면 스토리를 보게 됩니다. 여기에 Bum down 차트가 있는데, 여러분이 하기도 전에 뭔가를

읽을 수 있는 것과 같습니다. Bum down 차트를 따라서 완성된 요소들, 완성된 인위적인 요소들을 추적해나갈 것이고, 시간표에 따라 예측해볼 수 있습니다. 그리고, 여러분도 확실히 볼 수 있듯이 이 프로젝트는 11 월 1 일에 시작하고, 2010 년 6 월말에 끝나게 됩니다. 이제 이 프로젝트에서 정하고 싶은 몇가지 포인트를 갖게 됩니다. 100 개는 조금 안 되는군요. 이제 이 포인트들에 대해 살펴봅시다.

Scrum 의 각 역할에 대해서 알아보죠. 이 프리젠테이션에서 각 역할 들은 요구사항 관리에 맞게 적용했습니다. 첫째로 제품 소유자가 있구요. 제품 소유자는 제품 Backlog 를 소유하고 있습니다. 그래서 그는 backlog 의 모든 것들을 결정하고 우선순위를 매기며 이것이 가장 중요한 포인트입니다. 이 사람은 이 계획에 있어서 업무의 결과를 수용하거나 거절할 수 있어서 흥미로운 역할이며, 팀 내에 포함된 누군가가 될 수 있으며, 그들은 당신의 고객이 됩니다. 이 사람은 비즈니스에 속하는 사람을 불러올 수도 있는데, 이럴 때 풀타임으로 오게 되고 대부분 여러분의 고객입니다. 그리고 그들은 적당한 책임을 갖게 되고 권한을 갖습니다.

또 Scrum 마스터가 있는데, 그림은 캐나다 하키팀의 모습입니다. 저기에 C 마크가 써있는 셔츠 보이시죠? 모자에 그런 것이 쓰여져 있지만, 몇 분은 알아봤을 것 같습니다. 어떤 경우에도 Scrum 마스터는 프로젝트를 직접 관리하지 않지만 전체적인 Scrum 프로세스를 돕는 사람입니다. 이 사람은 프로젝트 팀이 Scrum 의 가치와 과정을 따라가는 지 관리하고 장애물을 제거하고, 또한 성공을 돕는 사람입니다. 이것이 프로젝트에서 프로세스를 처리하는 사람입니다.

그리고, 우리는 Scrum 팀이 있습니다. 일반적으로 Scrum 팀은 대개 4 명에서 10 명으로 구성되기 때문에 아주 작습니다. 10 명 이상이라면 아마도 여러분은 몇 개의 프로젝트를 조직하려 하겠죠. 중요한 건 각 기능별로 중첩된다는 겁니다. 여러분은 팀에 서로 다른 여러 구성원을 갖게 되고 또한 각 구성원들은 여러 가지 다른 역할을 수행해야 하죠. 어떤 프로젝트에도 풀 타임으로 지정되어야 합니다. 우리가 팀이 자기 스스로 조직한다고 할 때 누가 리더이고, 누가 선임, 후임인지 지정하지 않습니다. 팀이라고 할 때 여러분은 같은 목적을 향해 나아가는 것이고, 어떤 것이 우선인 지, 독립적으로 행해지는 업무가 합해지면 어떻게 되는 지 알고 있습니다. 그래서 여러분은 제품 소유자의 Scrum 마스터의 안내에 따라 조직을 꾸리게 됩니다.

이것들이 일부 역할들에 대한 것입니다. 이제 각 역할들에 대한 Scrum 의 영향에 대해 얘기해봅시다. 사실 프로젝트의 관점에서 보면, 조금 전에 살짝 얘기했지만 자기가 어떤 일을 할 지 선택할 기회가 있고, 이 것은 Sprint 을 계획할 때 나타납니다. 첫째는 상품의 소유주가 필요하다고 느낄 때, 우선순위를 재정렬할 수 있는 기회를 준다는 것입니다. 그래서 팀은 그들이 가장 높은 우선순위의 아이템을 선정하였는지를 확인할 수 있습니다. 그리고 독립성이나 기술적인 아이টে으로 인해 가끔은 그 우선순위가 비즈니스 우선순위와 관련이 없을 수도 있습니다. 그들은 높은 위험을 없애기 위해 노력하고 있으며, 상품 소유주와의 협상이 될 것입니다. 일주일에 한번 미팅을 하면서 프로젝트 현황을 확인하는 것은 Scrum 프로젝트에서는 일어나선 안 되는 일입니다. 매일 약 15 분 정도 미팅을 해야 하고, 짧다고 느끼겠지만 그래서 Stand-up 미팅이라고 합니다. 그리고, 3 가지 질문만 하는 겁니다. 내가 어제 한 일은, 내가 오늘 할 일은, 내 앞에 무엇이 있는가? 이 3 가지입니다. 이것이 Scrum 미팅에서 당신이 공유하는 것입니다.

이 사업을 공유하는 것은 매우 사업적으로 일어나는 것이고, 프로젝트 팀원들과 cart wall, burn down 차트를 공유하는 것입니다. Sprint 나 project 에 대한 Burn down 차트입니다. 이를 통해서 지금 이곳에서 일어나는 것이 쉽게 보이는 것이죠. 프로젝트 팀원들은 같은 목표를 이루려고 노력하기 때문에 자신들의 일이 끝나면 주변에 다른 할 일이 있는 지 둘러보게 됩니다. 만약 당신이 당신의 일을 끝냈다고 해서 다른 일을 하지 말고 Sprint 가 끝날 때까지 쉬세요. 당신이 도울 수 있는 뭔가를 발견한다면 성공적인 Sprint 를 위해서 그렇게 하세요, 모든 사람들이 자신의 일을 보여주게 되고, 매 sprint 가 끝날 때 자신의 일을 보여주게 되고, 당신은 그걸 바라겠죠. 당신이 한 일을 자랑스러워하고, 당신은 그걸 사용자들에게 보여주고 싶어하게 됩니다.

그것은 팀웍이 되고, 당신은 시험양식을 작성해줄 시험자도 갖게 될 수 있고, 앞으로 해야 할 일에 대한 분석을 해줄 수도 있죠. 고객은 사업적 결정을 내릴 권한을 갖고 있고, 이미 말했듯이 당신의 팀에 속해 있습니다. 흥미로운 포인트는 Scope 이 유동적인 것이 일정과 예산에 종속된다는 것입니다. 그래서 Scrum 프로젝트에서는 당신은 절대 범위가 있고, 이때까지는 이 일이 완료되어야 한다거나, 이 예산안에 일을 해야 한다거나 아니면 법정에서 봅시다 식의 말은 하면 안됩니다. 개념은 이런 겁니다.

여기 해야 할 일이 있습니다. 우리가 아마도 이 일을 끝내지 못할 수도 있다는 것을 알고, 또 우선순위가 바뀌기 때문에 업무를 다시 재배치해야 할 수도 있다는 것을 압니다. 고객이 요청하는 변경에 대응해야 할 수도 있고, 그것이 제품 소유자일 수도 있습니다. 그래서 우리는 또 제품 소유자가 권한이 막힐 경우 누군가 상위의 사람에게 이슈를 제기할 수도 있고, 프로젝트에 어떤 것을 제기하고 싶을 때 어떤 사람에게 언급을 해야할 지에 대해서도 알게 될 것입니다. 그래서 가능한 한 제품 소유자는 풀타임으로 프로젝트에 속해있습니다.

27:28 – 37:11

프로젝트 매니저, 저는 Scrum 에서는 프로젝트 매니저라는 역할이 있다고 생각하지는 않습니다. 어떻게 적용할 지에 대해 좀 흥미로운 것들이 있습니다. 팀원들이 더 높은 우선순위의 일을 선택하기 때문에 프로젝트 매니저는 통제권한이 줄어들게 됩니다. 상세한 계획이 없기 때문에 통상적인 프로젝트에서 봤던 PM 은 물속의 물고기와 같은 감정을 느끼게 됩니다. 당신은 상세한 계획을 짜는 게 아니고, 계획을 할 수 있는 것이라고 하면, 얼마나 많은 Sprint 를 갖게 될 지에 대해 계획을 세우는 것입니다.

이런 시간에 당신은 얼마만큼의 속도를 낼 수 있을까요? 프로젝트 중에 이만한 일들을 모두 다루는 것이 합리적일까요? Scrum 미팅에서 닭이 되고, 미팅에 참여한 누군가가 닭이거나, 아침에 달걀을 만드는 닭이 미팅에 참여했다고 생각해 보세요. 다른 편에 돼지가 프로젝트에 있는데, 프로젝트 매니저는 Scrum 미팅에서 그 닭입니다. 그들은 당신이 어제 뭘 했는지에 대한 상세한 현황을 원합니다. 그들은 사이트 미팅에서 필요한 것들을 펼쳐놓기를 원합니다. 그들은 팀을 도와주는 하인형 리더를 필요로 합니다. 팀은 이슈를 제거하는 것을 도와줍니다. 팀은 주간회의를 하지 않을 겁니다. 왜냐하면 그게 Scrum 이기 때문이죠. 그래서 우리는 Burn Down 차트를 갖고 있습니다.

Scrum 미팅에서 아주 가치적인 성과입니다. 속도는 정말 흥미로운 한가지인데, 우리는 항상 속도를 높이고자 하고 우리는 항상 옳지 않죠. 속도는 항목들의 개수이고, sprint 에서는 당신이 달성하고자 하는 만큼의 팀이 달성하는 업무의 양입니다. 첫 번째 Sprint 는 여러분이 예상한 것과 확실히 다른 숫자일 것이고, 다음 Sprint 는 조금 다를 것이고, 당신은 지난번 Sprint 의 속도와는 조금 다르게 계획을 세울 겁니다.

아마도 세 번째나 네 번째의 Sprint 에서 속도는 조금 낮게 시작할 것이지만 당신의 이제 잘 Sprint 를 계획할 수 있고, 우리는 많은 경험을 갖게 됐다고 얘기할 겁니다. 이제 우리는 적절하게 계획할 수 있죠. 이제 아마도 당신은 좀더 긴 계획단계로 넘어가고 프로젝트 매니저도 그럴 겁니다. 왜냐하면 우리는 프로젝트의 변경을 환영할 것이고, 예산을 다루어야 하기 때문에 변경을 관리해야 하기 때문입니다. 그래서 프로젝트 매니저가 익숙한 것보다는 많은 변경이 일어날 겁니다.

개발자들은 프로젝트에 좀더 빨리 합류할 겁니다. 제품 Backlog 에 대해 설명하고, 이해해야 하고, 얼마나 시간이 걸릴 지 예측해야 하기 위해서 그렇습니다. Sprint 를 계획하는 것을 도울 수도 있습니다. 그래서 개발자들은 통상적인 프로젝트에 비해 더 많이 참여할 수 있습니다. 일부는 제품 분석도 하고, 각 단계의 수행을 하고, 이를 마치면 다시 개발을 할 수도 있죠. 이 것들이 개발자들이 투입되는 때입니다. 테스트를 할 사람들도 Sprint 에서 개발되는 아이템들에 대해 도움을 주기 위해 일찍 참여할 겁니다. 물론 개발되는 동안이나 Sprint 에서 테스트가 가능할 때까지 기다릴 수도 있습니다.

그들은 여러 가지 역할을 해야 할 필요가 있기 때문에 다재 다능한 사람이 필요합니다. 사업분석가라는 역할을 여러분 중 얼마나 많은 분들이 해보셨는지 잘 모르겠지만, 우리는 사업 요구사항이라는 문서가 있고, 우리는 분석이라는 단계에서 이러한 것이 있었습니다. 사업분석가는 사업 요구사항 문서를 작성해야 하고, 이것은 고객의 승인을 받아야 하는 아주 중요한 일입니다. 처음부터 프로젝트 기간 동안 변경이 일어나지 않는 것은 Agile 형식이 아닙니다. 그래서 사업분석가는 Agile 에서 일어나는 것에 처음에는 놀랄 수 있는데, 그 것이 Sprint 의 backlog 입니다. 그리고 우리는 sprint backlog 에서 계획된 아이템에 따라 sprint 에서 여러 가지 것들을 이야기 시작합니다.

그리고 여러분은 아마도 Sprint 에 있어서 많은 자유를 가지게 될 것이며, 사업분석가뿐만 아니라 상세한 것을 필요로 하는 개발자에게도 마찬가지로 일 것입니다. 사업분석가는 프로젝트의 처음부터 끝까지 참여하게 되지만 그들은 그들이 익숙했던 형식의 요구사항이 아닌 것을 접하게 될 겁니다. 예를 들어 행복한 경로라는 것을 가질 수도 있는데, 또한 아주 낮은 레벨에서의 상세 사항에 있어서 변화된 것을 볼 수 있습니다. 이제 사업분석가는 우리 프로젝트팀원들과 아주 긴밀하게 일하게 되겠죠. 사용 사례들이 아마 처음부터 만들어 지지는 않겠지만, 어떤 사안이 정말 복잡하다고, 그것들을 만들어야 한다면, 우리는 사업분석가에게 당신의 도움이 필요해요 라고 요청할 것이고, 아주 복잡한 것이 있고, 우리는 이 것을 위해 사용할 필요가 있는 것으로 결정하였다라고 할 것입니다. 그리고 마지막으로 예측한다는 것은 상대적인 것입니다. 프로젝트의 한 아이템을 제품 Backlog 와 비교하는 것에 그렇죠. 그래서 우리는 어떤 모델이 가능하고, 얼마나 복잡한 것인지 결정지어야 합니다.

그래서 요구사항관리에 대한 영향은 사용 케이스를 쓰는 것에서부터 사용자 스토리까지 입니다. 우리는 별로 안좋아 보이는 사용자 스토리에 대한 몇 가지 간단한 사례를 살펴보았습니다. 정말로 사용자 스토리의 구조는 역할이어야 합니다. 어떤 때는 목표라고 하기도 하면서 달성하고 싶을 때, 그럴 때가 있어서 캠핑에 대한 예를 들고 싶습니다. 캠핑을 가고 싶어서 캠핑 자리를 예약하고 싶은데, 제가 도착했을 때 거기가 비어있기를 원하죠. 간단합니다. 이것이 서사시 인 건지 또는 기능인 지, 스토리의 집합인 지 아니면 실제 스토리일까요? 이 경우에 우리는 스토리라고 하는데, 너무 높은 레벨에서 시작하는 걸까요?

우리는 이걸 Sprint 에서 다루어야 하기 때문에 상세화할 필요가 있습니다. 하지만 여러분은 누구를 위한 것인지, 그들이 하고 싶은 게 뭔지, 왜 그들이 그걸 하고 싶어하는 지를 알아야 합니다. 그래서 항상 더 나은 케이스와 스토리, 데이터 다이어그램이 있고, 제가 말하고 싶은 것은 우리가 여기에 다루는 것처럼 아주 다양한 각도와 다양한 레벨로 스토리를 갖고 있다는 겁니다. 그리고 사실 여러분은 아주 낮은 레벨의 상세한 사항을 다루어야 하고, 아주 다양한 이유에 적합하도록 문서작성을 해야 할 필요가 있습니다. 그래야 사람들이 그 것을 이해할 수 있기 때문입니다. 그리고, 그 것은 지원을 받을 필요가 있기도 하기 때문일 것입니다. 하지만 대개 우리는 사용자 스토리로부터 시작하여 복잡도에 해당하는 만큼의 케이스를 사용합니다.

그래서 프로젝트 기간 내내 요구사항은 수집됩니다. 확실하게 말할 수 있음에도 불구하고, 전통적인 방식에서 요구사항을 수집하는 사업요구사항 문서는 Sprint 에서 꼭 작성되어야 한다고 하지는 않을 겁니다. 하지만, 요구사항 분석은 Sprint 전반에서 항상 일어나는 일입니다. 그 것은 시간적 분석에 있어서 기본적으로 여러분에게 기능, backlog 를 갖고 있고, 당신이 잘 모른다고 해도 Sprint 에서 스토리를 가지고 수행을 하다 보면 당신도 모르는 기회를 갖게 됩니다.

37:12 – 45:56

문서작성, 어떻게 이것들을 문서화할까요? 아마도 Sprint 에는 사업문서 계획은 없을 겁니다. 스토리를 통해서 이를 수행한다면, 여러분은 여기의 스토리와 같이 계층을 만들 수 있고, 서사시와 같은 것, 기능그룹, 스토리, 아직 상세화하지 않은 스토리 등이 있을 겁니다. 우리는 제품 backlog 를 갖고 있는데, 이건 사용자 스토리의 리스트죠. 이것들은 사업이나 제품 소유자에 의해 우선순위가 주어질 겁니다. 우선순위는 우리가 당연히 해야 하는 업무의 리스트인데, 이 리스트는 우리가 다루어야 하는 것에 대한 목록이 될 수 있고, 아마 우리가 하지 않을 것들의 리스트에 포함될 수 있습니다. 우리 중의 누구는 업무를 특정 짓고 싶어 할 수도 있고, 아닐 수도 있습니다.

프로젝트의 일부에서는 요구사항을 수집하고 정교화하는 사람들이 있는데, 이런 사람들을 사업분석가와 사업분석가의 영향이라고 할 수 있습니다. 진정으로 이러한 것들은 요구사항에 대한 일입니다. 요구사항을 관리할 때, 이것을 하는 사람들은 결국 요구사항 분석을 하는 팀원들이죠. 확실히 이걸 그들이 스토리에 대해 말하고, Sprint 의 끝에서 이것을 끝내고 싶어하기 때문입니다. 나는 정교화하고, 내가 그 것을 개발하기 위해서 내가 무엇을 알아야 하는지에 대해 알아낼 것이다, 이렇게 하는 거죠. 예상했던 것보다 복잡하면 분석가에게 도와달라고 하고, 적절하게 문서화할 것이다라고 하는 거죠. 아주 간단합니다. 회의록이 될 수도 있고, 어떤 스토리에 요구되는 문서의

케이스가 될 수도 있습니다. Scrum 의 협력과 의사소통에는 훨씬 더 많은 협력이 필요합니다. 모든 팀원이 매일 무얼 하고, 오늘 무얼 하고 내 앞길에 뭐가 있지? 하고 얘기할 겁니다. 모든 팀원은 Scrum 이후에 Scrum 마스터가 결정하는 것을 도울 수 있도록 이슈를 해결하는 것에 도움을 주어야 합니다.

많은 협력이 이루어 지고 있습니다, 일부는 아주 상당한 가시성이 있습니다. 여러분이 100 개의 아이템과 이름이 붙여진 카드가 있다고 해보죠, 여러분은 이 카드들이 시작부터는 아니지만 테스트할 준비가 될 때까지 움직이는 걸 볼 수 있을 겁니다. 그리고, 테스트를 위한 준비가 되었을 때부터 생산이 준비되었을 때 또는 결함이 식별된 때에까지도 그렇습니다. 통상적인 프로젝트에서는 보기 힘들지만 이런 협업과 의사소통은 아주 중요한 것입니다. 통상적인 프로젝트에서는 팀원들이 각자의 일에 집중하면서 큰 그림을 볼 수 없습니다.

필요한 문서에 대해서는 많이 언급된 것 같습니다. 다시 언급하지는 않겠지만, 요구사항을 문서화하는 것은 아까 카드로 다시 돌아가 보면 사람들은 물리적인 카드들이 붙여진 벽을 실제로 가지고 있습니다. 저는 몇몇 프로젝트에서 천장에서 바닥까지 카드들이 붙여진 벽도 보았고, SERA 라는 자동화된 제품을 이용해서 많은 카드들을 사용하는 것을 본 적이 있습니다. 이런 것의 장점은 구태여 다시 이 카드들을 재배열하려고 하지 않아도 된다는 겁니다. 자동화된 도구를 사용하면 많은 융통성을 갖게 됩니다. 만약 물리적인 카드 벽을 만든다면 아마도 문서화할 때 제약 사항들이 있을 겁니다.

상대적인 예측은 많은 사고를 요구하는 일입니다. 왜냐하면 우리는 몇 시간 몇 일이 소요되는 예측작업에 익숙하기 때문입니다 이제 Scrum 프로젝트에서도 이를 할 수 있지만, 그 것은 실제로 그렇게 해야하는 것이 아닙니다. 예측한다는 것은 상대적이고, 리스트의 다른 스토리와 비교해서 어려운 일입니다. 어떤 것은 예측이 안될 정도로 어렵기 때문에 아마도 여러분은 이걸 아주 크다고 할 수도 있습니다. 그래서 당신은 사람들이 곱셈리에 대해 하는 것처럼 얼마만큼의 곱셈리가 있고, 이 티셔츠 사이즈는 얼마지 하는 것처럼 얘기해야 합니다. 여러분이 숫자를 표시할 때 여기 보는 것처럼 지난 번 2 팀원의 합은 이것이고, 그래서 이 숫자는 이제 21 또는 13+8 처럼 표시해야 합니다. 이렇게 하면 여러분은 각 스토리를 다른 스토리와 비교할 수 있고, 그게 얼마나 큰 지 결정할 수 있습니다. 팀에게 업무를 맡길 때 확실히 Sprint 에서 다룰 수 있는 만큼 하는 거죠. 그래서 Sprint 의 속도를 설정하면 이제 적절하게 계획을 수립할 수 있습니다.

여러분은 우리가 Sprint 에서 36 개의 포인트를 짚어낼 수 있다고 말할지도 모릅니다. 그래서 여러분이 Sprint 계획을 시작할 때, 36 개의 포인트만 선택할 것입니다. 그리고 만약 다음 Sprint 를 위해 도전한다면, 그에 맞춰 조정할 것입니다. 스토리 포인트에서 중요한 한가지는 한 팀의 스토리는 다른 팀의 스토리와는 같지 않다는 겁니다. 그래서 그 프로젝트에 관여하는 팀에 맞게 매우 주관적입니다. 스토리 포인트를 수집하고 스토리 포인트에서 데이터베이스를 만드는 것에 저항하는 것은 이것이 당신이 예전 방식의 function point 를 만드는 것처럼 당신에게 유용하지 않기 때문입니다. 이걸 같은 얘기가 아니죠. 다른 한가지는 당신이 카드들이 있는 데크를 갖고 있을 때, 이것이 당신만의 상대적인 스토리 포인트일 수 있다는 겁니다. 그래서 다른 사람들이 아 이것은 해결됐고, 그러니 뒤집어두고, 이런 식으로 하다 보면 당신의 모든 카드가 뒤집어져 있을 수 있죠.

불일치하는 겁니다. 그래서 누군가는 2 개의 포인트라고 하고, 다른 사람은 13 개라고 했을 때, 대화를 통해 오해를 해결하고, 모두가 13 개 거나 아니면 2 개 거나 아니면 그 사이이거나 할 수 있도록 확실히 해야 한다는 겁니다.

45:57 – 50:44

정리하자면, Agile 팀원들은 다재 다능해야 합니다. 여러분은 각자 자기 프로젝트 업무를 하는 팀원들, 프로젝트의 다른 일도 할 수 있는 전문가 그리고 자기 일을 효율적으로 끝내고, 다른 팀원을 도와줄 수 있는 팀원들을 가져야 합니다. 다재다능하다는 것은 중요합니다. 할로윈파티를 생각해 보면 여러분의 팀원들은 이 가면으로 파티에 참석했다가 조금 후에는 다른 가면으로 변장해서 파티에 참석하는 식이죠. 결론은 Agile scrum 은 요구사항 관리에 영향을 주게 되고, 제가 누가, 무엇을, 언제, 어떻게의 오래된 방식에 따라서 요구사항관리에 영향을 주는 지 생각해봤습니다. 언제를 생각해 보면, 여러분은 비즈니스 요구사항들에서 높은 레벨의 사용자 스토리를 Sprint 에서 다루는 것으로 이동하는 겁니다.

이런 변화의 단계에서 많은 사람들이 문서화되어 있는 것을 정교화하는 데 어려움을 겪습니다. 하지만 그게 Agile 과 Scrum 이 의미하는 것입니다. 무엇을 생각해 보면, 높은 레벨 제품 Backlog 가 아주 중요해서, 요구사항이 어떻게 정의가 되고 어떤 인터뷰를 해서 비즈니스 문서가 만들어지게 되었는데, 이런 것이 모두 제품 Backlog 에 있습니다. 아마도 격식을 차리지 않은 형태일 수도 있고, 프로젝트 팀원에게 일어날 수도 있는 것이고, 아마도 제품 소유자가 당신을 위해서 분석하고 요구사항을 만들었을 수도 있습니다.

당신이 알아야 할 것에 대해 대답을 한다 라고 하거나 도와드린다고 하면, 아마도 어떤 레벨의 문서가 필요한 지를 파악하라는 것입니다. 당신이 해결하고자 하는 요구사항이 얼마만큼이나 복잡한 지입니다. 왜라고 하는 것은, 어떤 것이 앞으로 일어날 지, 높은 레벨에서 무엇을 해야 하는 지를 파악하고 그 이유를 찾는 것입니다. 누구라고 하는 것은, 분석가, 개발자, 시험자 등 협력해야 하는 모든 사람이죠.

scrum 에 대해 일반적인 것을 조금 더 알아보면, 여러분이 Scrum 을 써야 할 때를 생각해 보죠, 만약, 당신이 계약을 다루고 있고, 고정된 가격과 고정된 범위, 고정된 품질에 대해 말을 들었다면 당신은 정말로 고객과의 관계에서 신뢰를 확보해야 합니다. 일정이 고정되어 있다면 당신은 문제를 보게 될 것이고, scrum 프로젝트를 사용할 수 없습니다. 여러분은 아마도 제품 backlog 나 자기관리하는 팀 등에 대해 Scrum 접근법을 쓸 수도 있겠죠.

하지만, 당신이 융통성이 있는 제품 backlog 나 높은 비즈니스 요구사항을 맞추거나 변경에 대응 하는 등등 고정된 범위와 일정을 맞추는 데 어려움을 겪을 겁니다. 여러분의 고객이 충분한 권한을 갖고 있다면 제품 소유자를 당신의 팀에 끌어들이어야 합니다. 그렇지 않다면 많은 어려움을 겪게 될 겁니다. 당신은 제품 소유자가 당신의 팀에서 프로젝트를 돕게 해서, 우선순위를 부여하고, 제품 Backlog 에서 업무들을 빼는 것과 같이 당신을 돕게 해야 합니다. 이런 것이 당신이 요구사항을 관리하는데 영향을 미칠까요? 여러분이 생각하는 것을 말해보자면 당연히 Yes 겠죠. 프리젠테이션의

끝에 몇몇 링크를 작성해놓았습니다. 여러분들이 궁금해 하는 것이나 이 프리젠테이션의 링크입니다. 녹화가 될 것이고, 여러분이 이 프리젠테이션의 시간만큼 PDU 를 획득할 수 있을 겁니다.

50:45 – 1:00:45

질문: Agile 이 Spyro 와 SDLC 프로토타입과 상호작용하는 데 어떻게 다른가요?

답변: 이들은 상호 작용적입니다. 저는 Spyro 에 대해서는 잘 모릅니다만, 프로토타이핑은 무엇인가를 만들고, 고객에서 보여주는 것이라고 할 수 있는 개념입니다. 또 그것을 바꾸고, 다시 하고 그렇죠. 명확한 프로토타입을 생각해보죠, 그것은 어떤 일을 정의하고 우선순위를 나누고 하는 그런 것과는 조금 다릅니다. 고객은 자기가 원하는 것, 필요로 하는 것을 알고 있습니다.

상호작용의 접근법은 제가 언급한 scrum 과는 조금 다릅니다. 제가 얘기하는 것은 처음부터 당신은 Scrum 의 초점을 맞추고 거기에 집중할 것이고, 이 것은 모든 사람들이 당신이 Sprint 의 끝에서 완성을 할 수 있기를 기대합니다. 그들 모두 당신이 2 주나 아니면 당신이 선택한 일정 기간에 대해 알고 있습니다만, 그들은 또 바뀌지 않는 것에 대해서 알고 있습니다. 중요한 것은 업무와 기대를 설정하는 것이 확실히 가시성이 있어야 하는 것이고, 그 기대치에 맞추기 위해 개인이 아니라 팀으로 일을 해야 한다는 것입니다.

질문: Agile 로 전환할 때 어떤 사람이 힘들어할까요? 더 많은 Ownership 을 가지게 되는 개발자일까요? 통제권한을 잃게 되는 PM 일까요?

답변: 제 경험에서 얘기하면 개발자들은 좋아합니다. 왜냐하면 이래라저래라 하는 사람이 없어지기 때문입니다. 자신들이 하는 일에 대한 더 많은 통제권을 갖게 되기 때문이죠. 사업 쪽에도 그런 사람들이 있습니다. 제품소유자도 좋아합니다. 그러지 않은 사람이 PM 인데, 이 사람들은 여전히 2 년짜리 프로젝트에서 무엇이 일어나고 무엇이 일어날 지 상세한 계획을 알고 싶어 합니다. Agile 에서는 이런 일이 일어나지 않습니다. 그래서 여러분은 Agile 접근법에 대해서 PM 들을 훈련시켜야 합니다. 제가 경험에서 말씀드린다고 했는데, 제가 처음 Agile 에 대해 얘기했을 때, 저는 hybrid agile mythology 를 개발하는 팀에 있었습니다. 그때 경험으로 agile 에 대해 많은 것을 배웠고, 항상 그것만 생각하는 힘든 시간을 보냈습니다.

질문: artifact, 가공의 것들이 무엇입니까?

답변: 프리젠테이션에서 말씀드렸는데, 제품 backlog, 사용자의 스토리 등등입니다.

질문: 이상적인 Sprint 의 lifecycle 은 얼마간일까요?

답변: 프로젝트마다 다르겠지만, 기본적으로 agile 은 2 주에서 4 주입니다. 제가 말하고 싶은 것은 짧으면 짧을수록 좋다는 겁니다. 왜냐하면 당신은 제품을 납품해야 하고, 발표회는 무엇인가 잘못되었을 때 두 가지가 가까워질수록 서로간의 영향을 작아지게 됩니다. 예를 들면 6 개월짜리 Sprint 가 있다고 해보죠. 여기에는 Agile 프로그램이 없습니다, 왜냐하면 agile 이 아니고, 너무

단순하니까요. 뭔가가 계획에서 어긋난다고 하면 당신은 6개월 후에 발견하고 싶지는 않겠죠, 당신은 빨리 알아내야 하니까요. 그래서 짧으면 짧을수록 좋다는 것이지요.

질문: Sprint backlog 가 우선순위가 나뉘어나가요? Sprint 상자에 1 개 이상의 backlog 가 들어갈 수 있나요?

답변: yes 그리고 no 입니다. 제품 backlog 에는 우선순위가 있는데, 왜냐하면 당신은 계획단계에서 어떤 것을 선택해야 하기 때문이고, 다음 Sprint 에서 무엇을 해야 할 지 우선순위를 두어야 하기 때문입니다.

질문: 사업적인 역할이 전문적인 사업분석가나 Scrum 마스터 역할에 적합한 사람이 프로젝트 팀원 중에 있을 까요?

답변: 제품 소유자와 스토리를 잘 정성 들여 만들 수 있는 팀원이 좋겠죠. 이제는 무엇이든지 필요할 것입니다. 왜냐하면, 일반적인 역할에 따른 당신의 팀에서 사업 분석가라고 불리우는 팀원이 있기 때문이지만, 그가 아무런 역할을 하지 못하거나, 프로젝트팀원들 중에 사업분석을 잘 하거나 스토리를 잘 만들어내는 등의 역할을 잘 할 수 있는 사람이 있을 겁니다.

질문: Sprint 주기가 4 주가 넘어도 되나요? 아니면 4 주 단위로 끊어야 하나요?

답변: 이상적으로는 2 주에서 4 주라고 말씀드린 것 같은데요. 중요한 점은 Sprint 의 초점이 제품과 균형을 맞추는 것입니다. 무엇이 Sprint 에 있건 당신은 Sprint 의 초점을 바꾸지는 못합니다. Sprint 의 Backlog 중에서 아무것도 만들어내지 못한다면 그것은 프로젝트의 큰 문제가 될 것입니다. 프로젝트 팀이 Sprint backlog 에서 속도와 이슈들을 잘 활용한다면 목표를 달성할 수 있을 것입니다.

질문: 프로젝트 관리를 더 용이하게 해줄 수 있는 제안할 만한 도구가 있을 까요?

답변: 몇 가지가 있는데, mingle 이 그 한 가지구요. MS 의 여러 소프트웨어도 card wall 이나 task 지정, burndown 차트 등에 사용할 있는 tool 이 있습니다. 다른 여러 가지 tool 들도 있는데, 답을 드리자면 yes 라고 하고 싶구요. 그것들이 tool 입니다.

질문: 나열된 프로젝트의 여러 역할을 많은 사람들에게 공유하거나 서로 다른 역할에 같은 사람을 지정하는 것이 가능할까요?

답변: 다시 한번 말씀 드리지만, 여러분은 다른 여러 가지 역할을 할 수 있는 다재 다능한 사람을 갖는 것이 이상적입니다.

질문: scrum 마스터는 어느 단계에서 지정되어야 하나요?

답변: 빠르면 빠를수록 좋습니다. 당신이 Scrum 프로젝트를 시작할 때부터죠. 당신이 Scrum 마스터가 없다면 탈선할 위험에 처한 겁니다. 사람들은 뭘 하는 지 모릅니다 사람들은 자신들이 무엇을 하는지 알지 못하게 됩니다. Scrum 마스터는 가장 먼저 지정해야 합니다.

질문: 서로 다른 방법론에서의 각 역할에 대해 비교해주실 수 있나요?

답변: 잘 모르겠습니다. 제가 말하는 것은 우리는 폭포수 프레임워크에서 우리가 알고 있는 분석, 설계, 개발, 시험 등등과 비슷한 단계에 우리들이 익숙하다는 겁니다. 폭포수 프레임워크에서는 각 팀원들이 자기가 어떤 일을 하고 어떤 역할을 갖고 있는 지 알고 있습니다만 Scrum 프로젝트에서는 여러분은 다재 다능한 팀원을 원하게 될 겁니다.

질문: 우리가 같은 접근방식을 시간과 비용을 예측하는 데에도 사용할 수 있을까요?

답변: 당신이 원한다면 당신의 제품 Backlog 를 예측할 수 있습니다. 그것이 딱 Agile 한 것은 아닙니다만 Agile 은 좀 더 상대적인 예측방법이지만, 당신을 이 것을 통해서도 가능합니다. 사실 여러 프로젝트에서 그런 사례를 봤습니다. 문제는 어떤 주어진 시간에 완료될 것이라는 것에 대한 예측을 설정한 것입니다. 우리는 사실 우리가 몇 시간이 필요할 거다, 몇 주가 필요할 거다, 몇 달이 필요할 거다 등을 생각해서 말하는 것은 어리석은 일이라는 겁니다. 우리가 그냥 생각한 거다라고 인정하고, 우리를 도울 수 있는 상대적인 예측방법을 사용해보면 어떨까요?

이번 프리젠테이션에 참여해주셔서 감사합니다. 좋은 하루 되시구요, 다시 뵈 수 있기를 바랍니다.