



※ 본 아티클은 CMP MEDIA LLC와의 라이선스 계약에 의해 국문으로 제공됩니다

게임 개발 위키 길라잡이 (Learning the Ways of the Game Development Wiki)

팀 라이언 (Tim Ryan)
가마수트라 등록일(2009. 07. 30)

http://www.gamasutra.com/view/feature/4094/learning_the_ways_of_the_game_.php

[베테랑 게임 디자이너 라이언은 - 가장 최근 작품으로는 프랙처(Fracture)가 있다 - 위키(Wiki)를 하나의 게임 디자인 및 개발 툴로 보고, 질문을 던진다. 위키가 게임 제작 문서를 관리할 때 적합한 툴일까? 또, 대규모 게임들의 디자인 정보를 저장하는 데에 위키를 사용할 때의 장점과 단점에는 어떤 것들이 있을까?]

들어가기

자칭 올드스쿨 개발자인 나는 90년대 초반부터 게임 디자인 문서들을 직접 작성하거나 작성을 감독해왔는데, 최근 게임 디자인 스펙을 문서화하기 위해서 내부 위키 웹사이트를 사용하는 팀과 일할 기회가 있었다. 문서화 작업은 명확하고 정교해야 한다는 기준으로 볼 때, 처음에는 그다지 매력을 느끼지 못했으나 곧 이 작업방식의 여러가지 긍정적인 면들을 볼 수 있었다.

이리저리 돌아다니며 몇 가지 질문을 던지고 나자 꽤 많은 사람들이 위키를 활용하는 기업에서 근무한 것을 알 수 있었다. 그들 중 다수가 자체 출시할 게임을 개발하고 있었고, 따라서 퍼블리셔에 개발 중간보고서를 보내기 위해 문서를 인쇄 형식으로 만들어야 할 필요가 없었다.

나는 위키가 유용하게 사용될 수도 있을 것이라는 것을 깨닫고 익혀두기로 했다. 그래서 약간의 조사를 통해서 게임 디자인 문서화에 위키를 이용하는 것의 장점과 단점을 파악했고, 단점을 극복할 수 있는 방법들을 찾아냈다. 이 글을 통해 내가 알아낸 것들을 공유하고자 한다.

위키가 뭐야?

위키는 1994년 워드 커닝엄(Ward Cunningham)이 자기 회사의 프로그래머들이 아이디어를 공유할 수 있도록 발명했다. 이 간략한 마크업 언어는 흥미롭게도 웹 브라우저용 표준 마크업 언어 HTML의 선조로 알려진 애플의 하이퍼카드(Hypercard)를 개략적으로 본뜬 것이다. 위키는 HTML과 달리 훨씬 단순하며 오랜 세월 동안 통제 불가능한 복잡성 속에서 허우적대지 않았다. 누구나 편집 가능하다는 것이 핵심이다. 위키는 정보를 보여주기 위한 언어가 아니다. 정보를 공유하기 위한 언어이다. 이 언어는 높은 접근성을 통해서 커뮤니케이션과 그룹 기여도를 높인다.

십 수년간 위키는 다른 회사들로, 그리고 공공 웹사이트들로 퍼져나갔다. 그 중 가장 유명한 것은 모든 항목을 전세계의 개인들이 작성한 온라인 백과사전 위키피디아(Wikipedia)이다. 정보의

보고로서 신뢰받는 위키피디아의 명망은 위키의 목표가 성공했다는 것에 대한 궁극적인 증거이다. 워드에게 감사.

게임 개발에 위키가 적합할까?

지금은 커뮤니티와 상업용 사이트들에 사용되고 있지만, 위키는 원래 소프트웨어 개발 회사에서 사용하기 위해서 디자인되었다. 내가 조사한 바로는, 몇몇 게임 회사들은 이 언어를 꽤 오랫동안 사용해왔다. 점점 더 많은 비디오 게임 개발사들이 이 언어를 사용하기 시작하는 것은 전혀 놀라운 일이 아니다. 인기를 끌기 시작한 것이다.

우리 산업은 성장했다. 프로젝트의 규모, 게임이 다루고 있는 깊이, 그리고 정보의 절대량은 놀라울 지경이다. 한마디로, 디자이너나 제작자 혼자서 개발 문서를 작성하기에는 너무나 벅차다. 정보가 너무 많아서 섹션을 나눈다 해도 디자인 문서 하나에 담기 또한 벅차다. 위키는 여러 개인들이 작성할 수 있으며, 정보를 조직하고 검색하기 위해 브라우저 인터페이스를 사용하므로 이 문제를 해결해준다.

게임을 많이 하거나, 마이스페이스(MySpace)나 블로거(Blogger), 트위터(Twitter) 문화에 익숙해진 데에 따르는 부작용일지도 모르지만, 게임 개발자들은 대체로 집중시간이 짧고 위키가 장려하는 몇 바이트 단위의 조각글을 선호하는 것 같다. 그들은 100페이지짜리 디자인 스펙 문서를 뒤지는 것을 좋아하지 않는다. 위키 페이지는 각각의 주제를 가지고 있으며, 우리는 쉽게 이 주제를 다른 주제들과 링크하고, 섹션으로 묶고, 검색을 위해 레이블을 붙일 수 있다. 짧게 말해 위키는 디자인의 접근성을 좀 더 높여준다.

왜 셰어포인트를 쓰지 않고?

답은 간단하다.

1. 셰어포인트는 저렴하지 않다. 많은 위키 솔루션들은 무료이거나 상당히 저렴하다.
2. 셰어포인트는 거추장스럽고, 웹사이트라기보다는 문서 호스팅 솔루션에 가깝다. 웹 페이지처럼 웹 브라우저에 바로 뜨는 위키와 달리, 문서 하나를 사이트에서 받아 MS 오피스로 열어보기 위해서 문자 그대로 몇 분이나 기다려야 했다.
3. 많이 사용되는 툴은 그렇지 않은 툴보다 훨씬 유용하다. 내 경험상 팀원들은 위키를 더 손쉽게 사용했다. 더 많은 사람들이 글을 작성하고, 그에 더불어 더 많은 사람들이 읽는다.

위키 솔루션

여러가지 다양한 위키 솔루션들이 나와있다. 위키피디아에서 계속 추가되는 [리스트](#)를 찾아볼 수 있다.

위키 솔루션을 선택할 때 비용을 많이 고려한다면, 다양한 프리웨어들 중에서 선택할 수도 있다. 돈은 쓴 만큼 값을 한다는 사실만 명심하면 된다. 프리웨어들은 기본적인 기능만 제공하고, 고객 지원도 없다. 위키를 시험 삼아 사용해보는 거라면 이것들을 이리저리 만져보기를 추천한다. 하지만 실제로 사용하기로 마음먹은 순간, 선택에는 여러 기능들을 사용하고자 하는 욕구가 비용보다 더 큰 영향을 미칠 것이다.

주된 관심사는 얼마나 커스터마이징이 용이한지가 되어야 한다. 당신이나 동료 직원 중에 특정 프로그래밍 언어 - CSS나 PHP, 펄(Perl), 파이썬(Python), 자바(Java) - 에 능숙한 사람이 있다면 그 언어를 활용할 수 있는 솔루션을 선택하는 것이 유리할 수도 있다. 그러나 강력한 위키 솔루션일수록 프로그래머가 아닌 사람도 쉽게 사용할 수 있도록 플러그인과 매크로 편집창을 이용해서 기능을 추가할 수 있는 인터페이스를 제공한다. 당신이 원하는 그 기능을 제공해 줄지도 모를 수백 가지의 위키 플러그 인들과 매크로들이 오픈소스 협동의 정신에 입각하여 공개되어 있다.

이 글에서 내가 사용할 예제들은 티위키(TWiki)와 컨플루언스(Confluence)를 이용한다. 실제 위키 구문은 각 위키 솔루션마다 크게 다를 수 있다.

게임 디자인 문서화를 위한 위키 활용의 장점과 단점

아래의 장점과 단점들은 위키 활용과 기존의 워드 및 PDF 문서들을 비교한 것이다.

장점	단점
협동: <ul style="list-style-type: none"> ● 동시에 여러 작성자가 서로 다른 주제에 대해서 편집 가능 	산으로 가는 디자인: <ul style="list-style-type: none"> ● 벳사공이 많으면 디자인이 일관성을 잃고 편집자들은 서로 반목하게 된다
브라우저 친화적: <ul style="list-style-type: none"> ● 링크 페이지를 통한 분기 조직화 ● 검색 및 주제 찾기를 위해 레이블 사용 ● 외부 참고 자료에 대한 링크 용이 	끊임없는 변동사항: <ul style="list-style-type: none"> ● 즉시 공개, 어려운 버전 관리 ● 공식 결재 및 종료의 부재 ● 동적인 팀 목표
보안: <ul style="list-style-type: none"> ● 문서를 웹에 업로드하거나 이메일로 보내지 않음 ● 접근을 위해 도메인 보안을 사용 ● 그룹 및 개인별 접근 권한 관리 	낮은 휴대성: <ul style="list-style-type: none"> ● 인쇄용 포맷이 아님 ● 중간보고서 제출이 어려움 ● 처음부터 끝까지 읽도록 구성된 형식이 아님

장점: 협동

위키는 동시에 여러 작성자가 서로 다른 주제에 대해서 편집을 가능하게 한다. 최근의 기업들은 모두 이 트렌드를 따르고 있는 것으로 보인다. 극단적인 경우로, 여러 작성자들이 같은 스프레드 시트를 같은 시간에 편집할 수 있는 구글 닥스(Docs)가 있다.

마이크로소프트도 오피스에 비슷한 서비스를 추가하기 위해 개발 중인데, 조금 다른 방향을 추구

하고 있다. 현재로서는 네트워크에서 워드 문서를 탐색해서, 각자의 바탕화면에 띄운 후, 문단을 편집하고, “저장” 버튼을 누르면 다른 누군가가 그 파일을 사용 중이므로 수정할 수 없다는 메시지가 뜨는 것이 일반적인데, 위키를 사용할 때에는 작성자들이 정확히 같은 주제를 편집하고 있지 않는 이상 이런 문제가 발생하지 않는다.

단점: 산으로 가는 디자인

아마 다들 이런 속담을 들어봤을 테고, 어쩌면 직접 경험해 봤을 수도 있다: “사공이 많으면 배가 산으로 간다.” 확실히, 위키를 처음 접한 디자이너들 대부분은 그룹 디자인에서 발생하는 응집력의 부재를 가장 많이 걱정한다.

그들이 걱정하는 이유가 있다. 나는 그 동안 정도가 크든 작든 콘텐츠와 프리젠테이션에 대한 여러 편집자들 간의 반목으로 전혀 조직화 되어있지 않고, 일관성 없으며, 디자인 스펙 문서로서의 효용성이 없는 위키들을 봐 왔다. 모든 사람이 글을 잘 쓰는 것은 분명 아니며, 모두가 특별히 뛰어난 디자이너인 것도 아니다. 여기에 적용할 수 있는 속담이 또 하나 있다: “할 수 있다고 해서 꼭 해야 하는 것은 아니다. (Just because you can, doesn't mean you should.)”

내 경험에 따르면, 최고의 디자인 문서들은 기술문서 작성에 숙련된 디자이너들이 *귀를 열고* 작성했을 때 탄생했다. 그들은 다양한 의견들과 메모들, 그리고 팀에서 제시된 피드백들을 수령해서 팀의 목표와 아이디어를 대변하는, 응집력 있고 유용한 게임 디자인 스펙 문서를 만들어낼 수 있었다. 위키에는 수석 편집자의 개념이 존재하지 않는다. 위키 철학에 위배되기 때문이다.

위키 독재자가 되지 않고도 이러한 문제들을 피하기 위한 몇 가지 방법들이 있다.

1. 대부분의 위키들은 익명 사용자들이 수정할 수 없도록 설정할 수 있다.
2. 일부 위키들은 버전 관리 기능과 사용자 이름에 따른 수정 히스토리를 제공하므로, 바람직하지 않은 수정이 있었을 경우 책임 소재를 밝히고 대처하기가 용이하다.
3. 게다가, 대부분의 위키는 각 페이지마다 그룹이나 개인 작성자가 **읽기전용** 접근 설정을 할 수 있는 기능을 제공한다. 나는 사람들이 임베디드 데이터베이스 암호를 볼 수 없도록 할 때만 이 기능을 사용했다.
4. 각 콘텐츠 종류마다 스타일 가이드를 수립하고 모든 위키 페이지들을 카테고리화 하기 위한 프레임워크를 만든다.
5. 제시된 아이디어들과 메모, 그리고 개발 중인 스펙 문서들이 완성된 스펙 문서들과 분리되도록 공식적으로 위키 페이지들을 분류한다.
6. 위키 페이지들을 편집할 여유를 가진 수석 디자이너가 없다면, 위키 페이지들을 정리하고 조직화 하는 일을 하는 위키마스터(WikiMaster)를 별도로 임명한다.

장점: 브라우저 친화적

위키는 어떤 웹 브라우저에서든 작동한다. 위키 토픽 제목을 캐멀케이스(CamelCase)로 표기한

URL을 사용하므로 매우 쉽게 페이지를 찾을 수 있다. 모든 위키들은 검색을 위해서 페이지들에 태그를 달 수 있는 방법들을 제공한다 - 한마디로 쉽게 주제들끼리 엮거나 검색할 수 있도록 키워드를 부여하는 것이다. 컨플루언스에서는 이것을 *레이블*이라 부르고 티위키에서는 *태그*라고 부른다.

위키 페이지들은 상하관계로 그룹을 지을 수도 있다. 비교적 손쉽게 각 주제에 대한 네비게이션 트리를 가진 스위치보드 페이지를 만들 수 있다. 어떤 위키 솔루션을 사용하느냐에 따라서 네비게이션 트리는 자동으로 생성될 수도 있다.

위키는 하나의 위키 페이지와 다른 위키 페이지들, 첨부문서, 첨부파일들 또는 외부 참고 자료들과의 링크를 용이하게 해준다. 좋은 참고자료로 유튜브 동영상을 찾았다면 곧바로 위키 페이지에 추가할 수 있다. 워드 문서를 사용한다면 상상할 수 없는 일이다. 물론 기술적으로는 가능하지만, 인쇄가 불가능하고, 인터넷 연결에 의존하는데, 문서를 읽는 사람이 인터넷에 연결되어 있을지는 미지수다. 위키를 사용한다면 첨부한 참고 자료가 제대로 표시되리라고 확신할 수 있다.

단점: 끊임없는 변동사항

위키를 강력하게 만드는 특징들이 - 수정 속도와 접근성 - 끊임없이 디자인에 변동사항이 생기게 하는 요인이기도 하다.

그러면 팀의 목표가 고정되어 있지 않게 되어, 공식적인 리뷰와 종료 선언이 어려워지고, 디자인의 타당성이 전체적으로 의심받게 된다. 이를 극복하기 위한 몇 가지 방법이 있다:

1. 대부분의 위키들은 최근 수정된 페이지들을 보여주는 기능을 제공한다. 이 리스트를 모니터링해서 무엇이 수정되었는지 알아볼 수 있지만, 이렇게 하면 끊임없이 지켜보는 사람이 필요하고, 중대한 수정사항과 사소한 사항들을 구분할 수가 없다.
2. 일부 위키들은 편집이 완료되면 알려주는 기능을 제공한다. 편집자들은 코멘트를 달고 수정의 중대성을 수치화할 수 있다. 편집 알림은 각 주제를 읽기 위해 등록된 사용자들의 이메일을 통해서 보내진다.
3. 수정 결재를 관리하기 위한 다양한 플러그인들이 나와있다. 특정 사용자들이나 그룹들은 *결재* 지위를 부여 받을 수 있고, 몇몇 고급 플러그인들은 커스터마이징 가능한 결재과정 관리를 제공한다. 페이지를 수정하면 그 페이지에 대한 결재 과정이 시작된다.
4. 특정 레이블, 예를 들면 “Feature_Change” 레이블이 붙은 페이지들만 링크해 표시하는 위키 페이지를 생성할 수 있다. 그 이후로는 “Feature_Change” 레이블이 붙은 모든 페이지들이 그 위키 페이지에 링크되어 나타난다. 아래 예제를 보자:

티위키 편집모드 - 태그 검색 리스트

```
%SEARCH{ "Feature_Change" scope="all" type="keyword" nosearch="on" nototal="on" order="modified" reverse="on" format="| [[${topic}] | $wikiusername | $date |" limit="999" }%
```

티위키 렌더 모드 - 태그 검색 리스트

<u>This Topic</u>	UserB	06/18/09
<u>That Topic</u>	UserA	06/17/09
<u>Other Topic</u>	UserB	06/16/09

컨플루언스 편집 모드 - 레이블 네브맵

```
{navmap:Feature_Change|wrapAfter=3|cellWidth=120|cellHeight=120}
```

컨플루언스 렌더 모드 - 레이블 네브맵



컨플루언스 편집 모드 - 레이블에서 콘텐츠 추출

```
h1. Recent Feature Changes  
{contentbylabel:Feature_Change|showLabels=false|showSpace=false}
```

컨플루언스 렌더 모드 - 레이블에서 콘텐츠 추출

1. Recent Feature Changes

- This Topic
- That Topic
- Other Topic

5. MS 워드와 달리 위키에는 행 단위로 수정사항을 추적하고 코멘트를 다는 기능이 없다. 물론 색상을 넣은 코멘트를 추가할 수 있고, 수정된 섹션 부근에 색상 패널을 붙일 수 있지만, 그렇게 하면 위키 페이지는 색칠 공부 책처럼 보이게 되고, 나머지 위키 텍스트들과 코멘트들이 함께 내 보내기 된다. 이것들은 MS 워드에서처럼 화면 표시 방법을 바꾸거나 인쇄 옵션을 설정한다고 해서 사라지지 않는다.

나는 최근에 컴퍼지션(Composition)이라는 자바 플러그인의 텍스트 섹션 숨김 기능과, 한 사용자가 친히 인터넷에 공개한 “no-export” 자바 매크로가 적용된 그래픽 토글 버튼을 조합해서 문제를 해결했다.

나는 버튼 그래픽을 포스트잇처럼 보이는 것으로 바꿨다. 이렇게 하면 특정 행이나 표의 영역을 가리키는 포스트잇이 나타나고, 이것을 클릭하면 코멘트를 읽을 수 있다. 숨겨진 텍스트도 여전히 검색 가능하므로 원하는 구문을 추가할 수 있고, 수정사항은 검색 결과에 반영된다.

컨플루언스 편집 모드 - 태그 예제

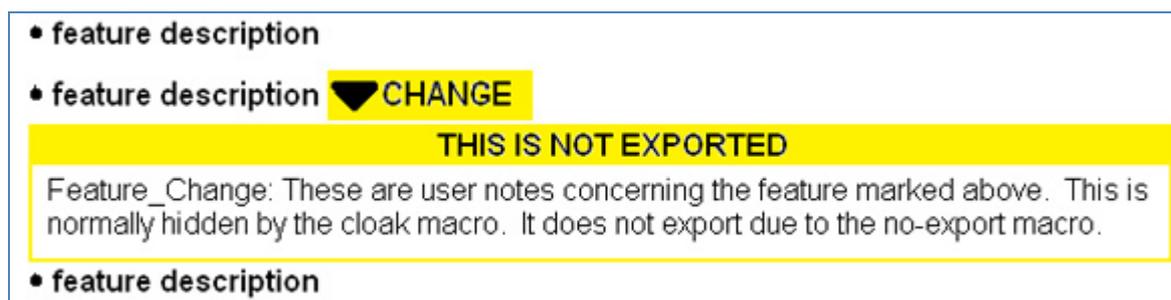
```
{composition-setup}
cloak.toggle.type=custom
cloak.toggle.open=http://gamecompanywiki/download/attachments/31075442/tag.gif
cloak.toggle.close=http://gamecompanywiki/download/attachments/31075442/tage.gif
{composition-setup}
```

```
* feature description
* feature description {toggle-cloak:id=*}
{cloak:id=*|visible=false}
{no-export}
Feature_Change: These are user notes concerning the feature marked above. This is normally hidden by the
cloak macro. It does not export due to the no-export macro.
{no-export}
{cloak}
* feature description
```

컨플루언스 렌더 모드 - 숨김 태그 예제



위키 렌더 모드 - 태그 클릭 후 숨김 해제



내보내기 안됨

Feature_Change: 이것들은 위에 표시된 기능과 관련 있는 작성자 메모입니다. 이 내용은 평상시 숨김 매크로로 숨겨져 있습니다. 이 내용은 no-export 매크로가 적용되어 있으므로 내보내기가 안됩니다.

장점: 보안

위키는 공개로 운영 할 수 있고, 비공개로도 회사의 도메인 보안을 이용해서 비공개로 운영할 수도 있다. 비공개 위키의 경우 사용자들은 도메인에 로그인 해야 내용을 읽을 수 있다. 이것으로 부족하다면 일부 위키에서는 등록된 사용자들에게 모든 권한을 부여하거나 읽기 전용으로만 접근하도록 할 수 있는 그룹 및 사용자 보안 설정을 제공한다.

접근 권한은 위키의 어떤 *네임스페이스(namespaces)*이나 특정 세부 항목에 대해서까지도 설정이 가능하다. 나는 양쪽 모두 설정하기를 추천한다. 이 경우 도메인 접근은 기밀 디자인을 볼 필요 없는 방문자들이나 원격 사용자들이 이용하게 되는 경우가 많다.

더 높은 보안 수준을 위해서 위키는 백업 기능을 제공하고, 우리는 이 기능을 이용해야 한다. 이용하는 솔루션에 따라, 위키 데이터는 단순한 텍스트 파일로 저장되거나 데이터베이스 형태로 존재한다. 시스템 관리자가 이 데이터들을 백업하도록 당부해 놓아야 한다. 데이터베이스가 꼬일 수도 있고, 사용자들이 위키를 망쳐버릴 수도 있으며, 외부로부터 공격을 받을 수도 있다. 이런 경우 테이프나 DVD 형태의 물리적 백업 수단이 구명줄이 될 수 있다.

단점: 낮은 휴대성

앞서 소개한 것과 같이, 위키는 인쇄용 포맷이 아니며 이메일이나 압축파일, 혹은 다른 어떤 방법으로도 워드 문서처럼 전송할 수 없다. 그러나 가장 진보된 위키들은 이런 문제들을 해결하기 위한 몇몇 옵션들을 제공한다. 사용중인 위키가 이런 옵션들을 제공하지 않는다고 해도, 이런 문제점들을 해결하기 위한 여러가지 솔루션들과 플러그인들이 나와있다.

1. 외부 인물과 그의 팀이 아무것도 손대지 않으리라는 신뢰를 가지고 있다면, 일부 회사들에서는 실용적인 해법으로 VPN 접속을 허용할 수 있다.

2. 일부 위키들은 HTML로 저장하기 옵션을 제공해서, 압축파일로나 FTP, 혹은 이메일로 전송할 수 있는 포터블 웹사이트를 만들어준다. 인쇄용 포맷은 아니지만, 온라인으로 주고 받을 수 있다.

3. 컨플루언스와 티위키 모두 PDF 형식으로 내보내기 옵션을 제공한다. PDF는 HTML처럼 링크가 살아있는 채로 온라인으로 주고 받을 수 있으며, 동시에 페이지로 나뉘어 있고, 목차가 존재하는 등의 포맷을 갖추고 있으므로 인쇄도 가능하다. 물론 완벽할 수는 없지만, 개발자가 특급 우편으로 퍼블리셔에 디자인 스펙 인쇄본을 보낼 수 있는 실용적인 옵션을 제공하는 것이다.

한가지 명심해야 할 것은 그래픽들은 페이지에 맞춰질 것이고 특정 페이지에 대한 가로 인쇄가 지원되지 않기 때문에 (적어도 내가 찾아본 바로는 그랬다) 표들이 레터 사이즈 그림으로 찌그러진 채 인쇄될 수도 있다는 점이다.

4. 어떤 내보내기 방법을 쓰건 간에, 언제나 그들이 읽어야 할 순서대로 위키 항목들을 조직화해

놓아야 한다. 표준 디자인 문서에서 형식을 따오고, 요약 페이지와 섹션 내 하위 페이지들로의 링크를 포함하는 섹션들을 생성하라. 그리고 새로운 주제가 올바른 섹션에 채워지도록 위키를 검토하라.

결론

처음에 나는 위키가 너무 방대하다고 생각했다. 너무 많은 페이지들과 작성자들로 인해 이 틀은 정보를 저장하고 있다가 보다는 혼란만 더하고 있는 것처럼 보였다. 내 워드 문서들이 그리웠다. 처음에는 모든 문서들을 표준 워드 문서 형식으로 작성한 다음 위키로 내보냈는데, 이것은 위키를 사용하는 목적에 부합하지 않는 방식이었다.

마침내 내게 광명이 찾아왔고, 나는 위키의 단점들을 피해서 작업하기 시작했다. 이 늙은 몸으로 새 재주들을 익혔다. 지금도 새 직원들을 위키에 입문시키면, 그들의 눈에서 나를 사로잡았던 것과 똑같은 혼란과 걱정을 읽을 수 있다.

그러나 오리엔테이션을 받고 나면 그들도 위키의 접근성과 팀 작업에 맞춰진 특성을 인정하게 된다. 바로 이 팀 작업에 맞춰진 위키의 특성이 프로젝트에서의 팀 공헌도를 높이고 일치된 의견을 이끌어 내는 것의 기반이 된다. 이제 이걸 사용하지 않고서 새로운 대규모 프로젝트를 진행하는 것은 꿈도 꾸지 않는다.

나는 티위키와 컨플루언스만을 사용해봤지만, 훨씬 더 많은 버전들이 나와있다. 틀림없이 여러분들 중 많은 수가 이전에 위키를 접해봤으리라 생각하는데, 사용해본 위키의 종류와 각자의 경험을 함께 나눠보자.