



※ 본 아티클은 CMP MEDIA LLC와의 라이선스 계약에 의해 국문으로 제공됩니다

## 신속하고 지저분한 시안 제작 성공담 (Quick and Dirty Prototyping: A Success Story)

파랄루먼 크루즈(Paraluman Cruz)

가마수트라 등록일(2010. 04. 01)

[http://www.gamasutra.com/view/feature/4314/quick\\_and\\_dirty\\_prototyping\\_a\\_.php](http://www.gamasutra.com/view/feature/4314/quick_and_dirty_prototyping_a_.php)

지난 11 월 IDGA 리더십 포럼에서 필자는 소형 PC 게임을 제작하는 한 방식으로서의 ‘신속하고 지저분한 시안 제작’에 대해 짚막한 발언을 한 바 있다. 대부분의 제작자들이 기존의 워터폴 내지 애자일 방법론(waterfall or agile methodologies)에 이미 안주하고 있어서인지 필자의 발언에 신기하다는 반응이었다.

필자의 스튜디오인 붐잡 엔터테인먼트(Boomzap Entertainment)는 글자 그대로의 의미로서 애자일(agile)하지만(신속하고 유연하다는 의미) 스크럼이나 XP 같이 인기 프레임워크를 따르지는 않는다. 대신 지난 5 년 동안 우리 같은 인디 스튜디오에게 가장 잘 맞는 프로세스를 만들어냈다. 필자는 이 프로세스를 ‘신속하고 지저분한 시안 제작’이라고 부르고 싶다. 하지만 필자 말고 붐잡 안에서 이를 그렇게 부르는 사람은 없다.

그 포럼에서 발표했던 것을 한 가마수트라 기사에서 짚막하게 소개한 바 있다. 그에 이어지는 이 특집 기사에서 필자는 우리 스튜디오가 ‘신속하고, 지저분하게 시안을 제작하는 방법’, 사용하는 툴과 기법, 그리고 이 방식의 장점(및 위험)에 대해 심도 있게 논하고자 한다. 이를 위해 우리의 최신 게임인 ‘Awakening: The Dreamless Castle’을 사례로서 이용할 것이다.

### 이게 효과가 있는 이유

본론으로 들어가기 앞서 이해를 돕기 위해 우리 회사를 간단히 소개하자면 붐잡 엔터테인먼트(Boomzap Entertainment)는 동남아시아에 있는 소규모 인디 캐주얼 게임 개발자이다. 100% 가상 스튜디오로서 말레이시아, 싱가포르, 일본, 필리핀에 걸쳐 있고 모두가 재택 근무를 하고 임대 사무 공간이 따로 없다.

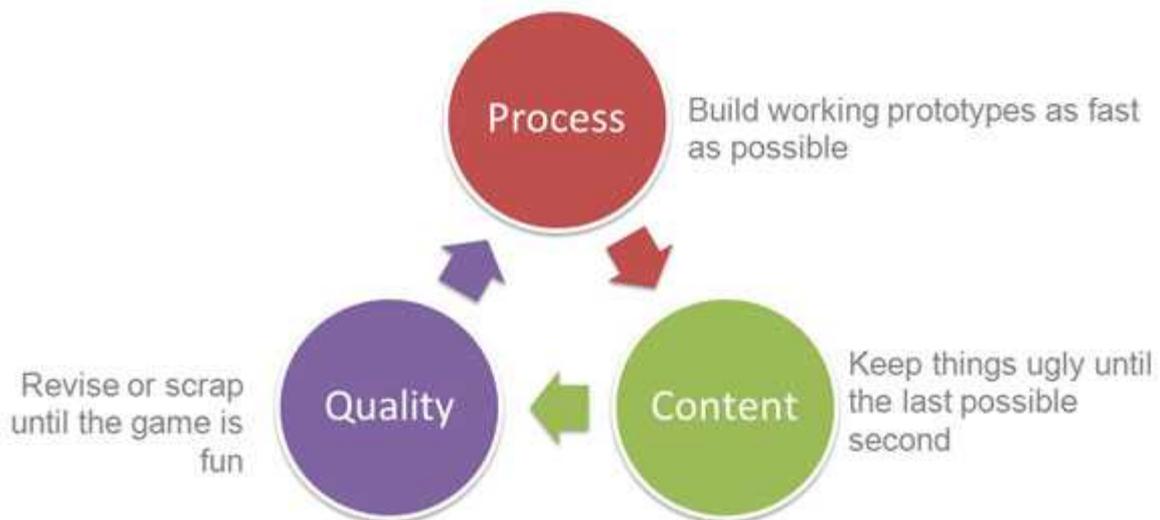
‘Awakening’ 제작팀은 회의 및 문서화에서 애셋 및 빌드 프로세싱에 이르기까지 순전히 온라인을 통해서만 함께 일했다. 우리는 오로지 다른 사람이 빌드에 집어넣는 작업물만을 보면서 이들의 능력을 판단할 수 있었고 이러한 작업물을 통해 우리는 시안 작업에 기여하게 된다(본질상 결과지상주의에 해당한다).

우리 회사는 작다. 고작 19 명이 전부이다. 팀은 프로그래머 1 인, 디자이너 1 인, 아티스트 2-3 인, 이렇게 4-5 인으로 이루어지는 게 보통이다. 'Awakening' 팀에는 최대 7 명까지 참여한 적이 있었는데, 아트 위주의 게임이어서 태반이 아티스트였다.

우리 팀은 인원이 적어서 어떤 위계나 관료적 제도 같은 게 전혀 필요치 않았다. 누구라도 아이디어를 개선할 수 있었고, 또한 이를 승인하고 서로 동등하게 투표를 했다. 또한 누구든지 신속히 무언가를 시안으로 만들어 빌드로 보낼 수 있었고, 팀의 나머지 사람들은 이런 일을 갖고 특별히 뭐라 하지 않았다.

결과만이 전부인 가상 환경과 캐주얼 게임을 만드는 소규모 팀이라는 조합에 의해 신속하고 지지분한 시안 작업이 수월하게 이뤄졌다. 그렇다고 해서 이게 대형 스튜디오나 프로젝트, 팀에는 효과적이지 않다는 말이 아니다. 당신의 스튜디오 역시 어쩌면 이와 비슷한 것을 하고 있을 지도 모른다.

## Quick and dirty prototyping



### 신속하고 지지분한 시안 작업

	프로세스		최대한 신속히 실행 가능한 시안을 제작
게임이 재미있어질 때까지 수정하거나 버리기	품질	컨텐츠	최대한 늦게까지 조잡하게 유지

### 작용 원리

'신속하고 지지분한 시안 작업'은 3 가지 기본적 생각들을 바탕으로 한다.

1. 최대한 신속히 실행 가능한 시안을 제작한다.
2. 최대한 늦게까지 애셋들을 조잡하게 방치한다.
3. 재미 있을 때까지 컨텐츠를 수정하거나 버린다.

오로지 속도와 효율에만 전적으로 치중하는 방식이다. 'Awakening'을 비롯한 우리의 모든 프로젝트에서 일일 빌드와 플레이스홀더 애셋, 신속한 이터레이션은 하나의 정해진 일과로서 자리잡았다.

### 단계 1: 실행 가능한 시안을 최대한 빨리 만들자

우리는 데모 개발 첫날부터 출시까지 'Awakening'의 일일 빌드를 성실히 만들었다. 빌드는 내부 프로젝트 사이트와 퍼블리셔의 사이트에 올려졌다. 따라서 퍼블리셔는 어느 빌드라도 서버에서 불러와 실행해볼 수 있었다. 심지어 불량 빌드마저 그렇게 할 수 있었다. 빌드는 항상 변경 로그와 함께 올려져서 퍼블리셔와 QA 가 별다른 어려움 없이 매일 어느 룬과 퍼즐을 테스트해야 할 지 알 수 있게 했다.

일일 빌드 제작의 가장 큰 장점은 아이디어를 신속히 테스트할 수 있다는 점이다. 누군가가 새로운 아트 애셋이나 부분게임(subgame)을 게임에 집어 넣고 싶어 한다면 우리는 '그것을 빌드로 올려 어떤지 한번 보자'고 말하곤 한다. 이는 말하자면 행궁 및 반복(rinse-and-repeat: 목표에 도달할 때까지 실행을 반복하는 것) 프로세스이다. 어떤 아이디어를 빌드로 올리면 다음날 이를 테스트하고 신속히 피드백을 받는다.

우리는 가상 회사였으므로 팀 구성원은 자신의 작업을 스스로 빌드로 올려야 했다. 이의 원활한 진행을 위해 몇 가지 중요 도구를 이용했다. 팀원들은 이메일(중앙 이메일 주소 1 개)이나 소스 컨트롤(서브버전(SVN) 이용)을 통해 애셋을 제출했다.

하루가 끝날 때 프로그래머는 배치파일을 실행해 SVN 과 이메일에 있는 모든 파일을 수거하여 프로세싱이 필요한 것을 변환 내지 내보내기하고, 코드를 컴파일했다. 이 과정을 최대한 자동화함으로써 인간적 오류를 최소화했다.

애셋을 스크립트하고 인코딩하는 작업에는 주로 MS 엑셀을 이용했다. 엑셀 안에 집어넣을 수만 있다면 변수, 게임플레이 스크립트, 애셋 속성, 음향 효과, 문자열 할 것 없이 뭐든 집어 넣었다. 이러한 것들을 코드 밖에서 유지함으로써 프로그래머의 지원 없이도 값을 손쉽게 조정하고 빌드에서 테스트할 수 있었다.

SpriteName	Filename	SpriteC	FPS	BorderX	BorderY	FlipU	FlipV	ClipX	ClipY
icon_crowbar001_lg	img/icon_crowbar001_lg.png	f	f	0	0	0	0	0	0
icon_crowbar001_sm	img/icon_crowbar001_sm.png	f	f	0	0	0	0	0	0
icon_ho024_item_001_lg	img/icon_ho024_item_001_lg.png	f	f	0	0	0	0	0	0
icon_ho024_item_001_sm	img/icon_ho024_item_001_sm.png	f	f	0	0	0	0	0	0
icon_insignia001_lg	img/icon_insignia001_lg.png	f	f	0	0	0	0	38	38
icon_insignia001_sm	img/icon_insignia001_sm.png	f	f	0	0	0	0	0	0
icon_insignia002_lg	img/icon_insignia002_lg.png	f	f	0	0	0	0	38	38
icon_insignia002_sm	img/icon_insignia002_sm.png	f	f	0	0	0	0	0	0
icon_insignia003_lg	img/icon_insignia003_lg.png	f	f	0	0	0	0	38	38
icon_insignia003_sm	img/icon_insignia003_sm.png	f	f	0	0	0	0	0	0
icon_insignia004_lg	img/icon_insignia004_lg.png	f	f	0	0	0	0	38	38
icon_insignia004_sm	img/icon_insignia004_sm.png	f	f	0	0	0	0	0	0
icon_key001_lg	img/icon_key001_lg.png	f	f	0	0	0	0	28	28
icon_key001_sm	img/icon_key001_sm.png	f	f	0	0	0	0	0	0
icon_key002_lg	img/icon_key002_lg.png	f	f	0	0	0	0	28	28
icon_key002_sm	img/icon_key002_sm.png	f	f	0	0	0	0	0	0
icon_rake001_lg	img/icon_rake001_lg.png	f	f	0	0	0	0	0	0
icon_rake001_sm	img/icon_rake001_sm.png	f	f	0	0	0	0	0	0
icon_m001_item_008_lg	img/icon_m001_item_008_lg.png	f	f	0	0	0	0	43	56
icon_m001_item_008_sm	img/icon_m001_item_008_sm.png	f	f	0	0	0	0	8	17
icon_m001_item_009_lg	img/icon_m001_item_009_lg.png	f	f	0	0	0	0	50	51
icon_m001_item_009_sm	img/icon_m001_item_009_sm.png	f	f	0	0	0	0	19	22
icon_m001_item_011_lg	img/icon_m001_item_011_lg.png	f	f	0	0	0	0	12	12
icon_m001_item_011_sm	img/icon_m001_item_011_sm.png	f	f	0	0	0	0	29	31
icon_m001_item_017_lg	img/icon_m001_item_017_lg.png	f	f	0	0	0	0	2	4
icon_m001_item_017_sm	img/icon_m001_item_017_sm.png	f	f	0	0	0	0	13	16

시트는 사용하기 쉽게 만들었다. 각 스프레드시트의 최상위 열에는 커다란 '내보내기' 버튼이 자리했다. 이를 클릭하면 데이터를 엔진이 읽을 수 있는 .lua 파일로 내보내기하는 매크로가 실행된다. 아티스트와 디자이너가 데이터와 애셋을 스스로 처리함으로써 프로그래머는 마음껏 자신의 직무와 코드 작업을 할 수 있었다. 누구도 엑셀로부터 벗어날 수 없다. 아티스트조차 중요 시간대에는 스프레드시트에 딱 달라붙어 있었다.

좋지 않은 점은 누구나 무엇이든 빌드로 올릴 수 있기 때문에 사고는 어찌됐든 발생하게 돼 있다는 점이다. 고의든 그렇지 않든 누구나 실수를 하게 마련이니까 말이다.

이를 피하고자 각 단계들을 이러한 실수로부터 최대한 안전하게 만들었다. 여기에는 소스 컨트롤이 주효했다. 누군가가 망가진 애셋이나 작업시트를 올리더라도 신속히 이전 작업 버전으로 되돌아갈 수 있었다(누가 무엇을 올렸는지 파악할 수 있도록 로그를 유지했지만 보복적이 되지 않도록 노력했다.).

아울러 누가 됐든 작업물을 제출하기 전에 스스로 테스트를 하도록 했고 프로그래머도 빌드를 업로드하기 전에 자체적으로 테스트를 했다. 테스트 작업을 소홀히 했던 날 공교롭게도 퍼블리셔가 그날의 빌드를 가져간 경우가 몇 차례 있었다. 이를 계기로 어쩔 수 없이 테스트를 계속해야만 했다.

## 단계 2: 애셋을 최후의 순간까지 조잡하게 유지하자

일일 빌드 프로세스는 적절하고 좋아 보이지만 현실적으로 말해 하루에 할 수 있는 작업이란 게 도대체 얼마나 될까? 아트 애셋은 캐주얼 게임에서조차 완성하기까지 며칠은 걸리는 경우가 허다하다(예컨대 Awakening 에서 '히든 오브젝트(Hidden Object)' 장면은 아티스트 1 인이 처음부터 맡아 하는 경우 최대 1 주일이 걸릴 수 있다).

하지만 신속하고 지저분한 시안 작업에서 1 주일은 용납이 안 된다. 해당 아트가 우리가 원하는 게 아니라면 어떻게 해야 하나? 어떤 기능이 하나도 재미가 없다면 어떻게 될까? 꾸준히 일일 빌드를 만들 수 있는 유일한 길은 작업을 가급적 간소하고 '가볍게' 유지하는 것이었다. 다시 말해 최후의 순간까지 애셋을 조잡하게 유지했다.

Awakening 의 대부분에서 항상 플레이스홀더를 이용했다. 아트는 말할 것도 없고 플레이스홀더 효과, 음향, 텍스트 역시 이용했다. 우리의 첫 번째 이정표는 전체 게임에 대한 실행 가능한 흑백 빌드를 만드는 것이었다. 처음부터 끝까지 성 전체를 돌아다닐 수 있었고 오브젝트를 클릭하면 흑백의 클로즈업이 나타나게 할 수 있었고, 아이템을 선택해 인벤토리 내에 나타나도록 할 수 있었고(조잡한 클립아트 물음표로서), 아울러 지극히 기본적인 부분게임을 실행할 수 있었다.



여기서 말하는 흑백 아트란 아티스트가 정말 스케치한 라인 아트가 아니다. 대다수가 조잡한 초벌 그림에 불과했다. 환상 속의 세상이 그와 같이 보인다고, 그리고 이를 퍼블리셔에게 하나의 이정표로 제출한다고 생각해보라.

다행히 퍼블리셔는 우리가 일하는 방식을 알고 있었고, 게다가 이의 가치를 제대로 알고 있었다. 덕분에 퍼블리셔는 게임을 매우 이른 시기에 실행해볼 수 있었고, 게임 세계가 어느 정도 크기인지, 퍼즐이 그럴 듯한지 그리고 부분게임들이 재미있는지 판단할 수 있었다. 엉터리

아트에 두리몽실한 효과, 가짜 텍스트를 갖고 게임을 어떻게 실행할 수 있는지 의아해할 지 모르겠다. 하지만 약간의 상상력만 있으면 충분하다.



우리는 플레이스홀더를 이용함으로써 아트가 최종적으로 마무리되기를 기다림 없이 우리 일을 할 수 있었다. 실제 아트가 완성되기를 기다림 없이 오브젝트 이터레이션을 스크립트할 수 있었다. 그게 아니라면 아티스트는 진행 중인 작업을 빌드로 보내 즉각적인 피드백을 받을 수 있었다. 의존성을 최소화하는 것이 가상 오피스 환경에서 극히 중요하다. 아무 것도 하지 않으면서 그냥 앉아 있으려는(무언가가 온라인에 올라오기를 기다리면서) 유혹이 정말 강한 환경이다. 하지만 우리는 이를 용납하지 않는다. 플레이스홀더만 있으면 대부분의 작업을 할 수 있다. 하다못해 조잡한 초벌 스케치라도 말이다.

### 단계 3: 게임이 재미 있어질 때까지 콘텐츠를 수정하거나 버리자

일일 빌드와 플레이스홀더 애셋에 의해 매우 신속한 변경이 가능했다. 이게 '신속하고 지저분한 시안' 작업의 핵심이다. 게임 내의 무언가에 대해 마음이 바뀌더라도 변경의 대가는 전통적 환경에서보다 훨씬 더 적다. 대부분의 경우 단지 플레이스홀더나 며칠 동안의 작업물만 다뤄주면 됐다.

알파 버전이 완성될 무렵 아트가 최종적으로 완성됐다. 그리고 이때쯤 대부분의 기능에 대한 실행 테스트를 끝마쳤고, 이렇다 할 대가 없이 수도 없이 변경을 했다.

우리는 매일 빌드를 만들기 때문에 테스트 역시 매일 이뤄져야 했다. 매일 아침에 일일 빌드를 다운로드해 테스트를 했다. 우리를 2 가지 사항을 확인한다. 첫째, 빌드가 효과가 있나?(즉, 우리가 퍼블리셔에게 말한 대로 되었는가?) 둘째, 이는 더 중요한 것인데, 빌드가 재미있었나? 지난 밤 올린 새로운 부분게임이나 애셋이 실제로 실행하기에 재미있냐는 것이다.

‘재미’는 언제나 주관적이고, 우리 팀은 너무 깊숙이 몰입해 있어서 투명한 판단을 내릴 수 없었다. 그래서 이따금씩 회사 차원의 게임 실행 테스트 자리를 마련해 친척과 친구들을 끌어들었다(가끔 퍼블리셔들도 참여시켰다).

프로젝트에 관여한 적이 없는 사람들이 게임을 보고 신선한 의견을 제시하는 것은 크게 도움이 됐다. 우리는 회사 내 나머지 사람들을 크게 방해하지 않으려고 노력했고, 임박한 이정표 전에만 대대적인 게임 실행 테스트 자리를 준비했다.

게임 실행 테스트는 온갖 곳으로부터의 기나긴 주석 리스트를 만들어내곤 했다. 이 주석 리스트는 일일 빌드와 함께 더 길어진다. 집중력을 유지하고 일정에 차질이 없도록 하기 위해 단일 중앙 작업표에 따라 일을 했다. 제반 버그 및 변경 요청은 스프레드시트에 기재하여 1 - 4 까지 우선 순위가 매겨졌다(1 은 중대함을 의미). Awakening 의 제작 내내 이 작업표를 MS 엑셀에서 구글 독스로 옮겨 실시간으로 볼 수 있도록 했다.



작업들이 적절한 테스트 없이 무작정 배제 내지 삭제되지 않도록 작업표 소유자(대개 크리에이티브 디렉터나 프로듀서)를 두었는데, 그만큼 어떤 작업이 완료된 것으로 표시할 수 있었다. Awakening 의 작업표는 딱딱한 격식 없이 유지되었고 주석은 최대한 짧게 만들었다. 여기서 다시 ‘신속하고 지저분하면서도 효과적으로’가 목표가 되었다.

알파와 베타 중에 작업표의 기재 항목은 심각한 충돌 버그에서부터 고블린종 박스의 목재 색상에 대한 제안에 이르기까지 1,000 개가 넘어갔다. 가장 중요한 작업에 집중하기 위해 우선순위가 3-4 등급(우선 순위가 가장 낮음)에 속하는 작업은 모두 숨겨지도록 했고, 다른 스프레드시트에 배치했다.



이런 식으로 중요한 작업을 우선적으로 처리했다. 이게 훨씬 더 어렵더라도 말이다. 누군가가 우선순위 등급 4 에 해당하는 자신의 코멘트가 실행되지 않았다고 불쾌해하면, 곧 그렇게 할 테니까 그 정도로 해두라고 농담을 하곤 했다.

### 실제로 해보기

Awakening 은 우리가 만든 첫 번째 캐주얼 어드벤처 게임이었고, 신속하고 지지분한 시안 작업은 설계, 아트 스타일, 툴을 마무리하는데 큰 도움이 되었다. 이 프로젝트는 1 레벨짜리 데모로 시작했고 이는 약 3 주에 걸쳐 완성됐다. 데모는 완전한 오브젝트, 서브게임, 효과를 가진 가든룸과 하나의 히든 오브젝트 장면으로 구성됐다.

데모를 만든 가장 큰 이유는 퍼블리셔를 물색하기 위함이었으므로 이때만큼은 ‘조잡하게 유지하자’는 규칙을 지키지 않았다. 가든 데모는 매우 깔끔했는데 Awakening 의 최종판과 비교해 80%정도의 품질이었다. 짧은 시간 내에 게임의 완전한 버티컬 슬라이스(vertical slice: 시연 내지 테스트용의 종적 샘플)를 완성하는 일은 팀의 능력을 가능하는데 도움이 됐다. 그럴 능력만 된다면 주저 없이 뛰어들 것이다.

일단 계약을 마치고 완성까지는 10 개월 정도가 소요됐다. 거의 알파까지 대부분의 아트가 플레이스홀더로 유지됐다. 우선 흑백 스케치로 성 전체를 구성했는데 이때는 아트 품질이 50%

정도였고(대충 실행 가능한 수준), 그 후 알파 버전에서 80%로 끌어올렸다. 마지막으로 시각 효과와 음향, 컷신을 추가했다.

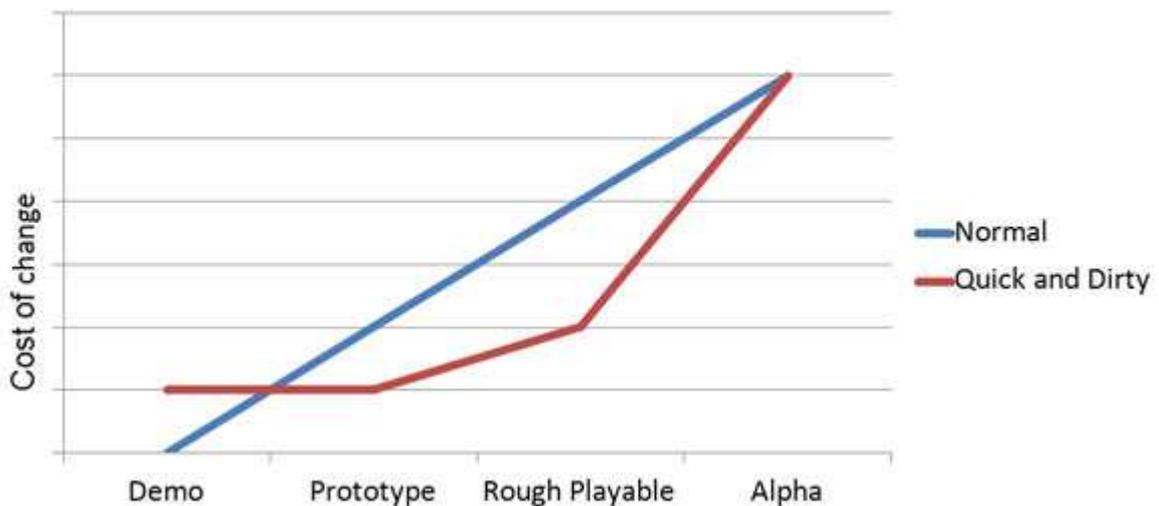
최후까지 유연성을 유지함으로써 개발 마지막 달에도 변경을 할 수 있어서 게임의 전반적 실행성에 참으로 도움이 됐다.

일일 시안 작업은 애셋이 주는 수많은 속박으로부터 우리를 자유롭게 했다. 우리는 게임의 주요 부분들을 내버리거나 변경했다. 별로 나쁘지 않았는데도 말이다. Awakening 은 빅피쉬게임즈에서 거의 3 주 동안 1 위 자리를 차지했다. 필자는 ‘신속하고 지저분한 시안작업’이 주는 유연성 때문에 이러한 성공이 가능했다고 믿는다.

낭비 대 가치  
 변경에 따른 대가  
 정상  
 신속하고 지저분함

데모	시안	대충 실행 가능	알파
----	----	----------	----

## Waste vs. value



### 성공적인 시안작업을 위한 팁

‘신속하고 지저분함’ 기법의 진정한 의미는 엄격하거나 정착된 어떠한 규칙도 없다는 것이다. 우리는 앞으로 나아가면서 프로세스들을 팀에 가장 유익한 방향으로 조정했다. 다른 스튜디오를 위해 우리가 나누어줄 수 있는 포괄적인 지혜가 있다면 이는 다음과 같다.

1) 팀이 시안 관점에서 사고하도록 훈련시키자. 아이디어를 실험하도록 장려하고 아이디어가 실패했을 때 이를 받아들이도록 장려하자. 아티스트는 작업물을 얼마든지 만들고 제출할 수

있도록 하자. 그리고 완벽함은 최종 아이디어를 위해 남겨두자. 누구나 틀을 능숙하게 다룰 수 있도록 훈련시켜 아이디어를 스스로 시안으로 만들고 테스트할 수 있도록 하자.

1) 퍼블리셔를 훈련시키자. 현재 진행 중인 작업이 어느 부분인지 설명하고 프로세스의 이른 시기부터 피드백에 끌어들인다. 이들의 참여를 장려하지만 변화가 적당하지 않은 경우가 언제인지도 알게 한다. 변경을 신속히 이행할 수 있다고 해서 이게 전혀 시간이 걸리지 않고 부담이 없다는 의미는 아니기 때문이다.

1) 수정을 중단해야 할 때가 언제인지 배우자. 이정표 일자들은 변경 불능이라 간주하고 어떤 기능이나 애셋이 일단 승인되면 확고한 태도를 취하자. 마음을 매일 변화시킬 수 있다고 해서 실제로 마음을 매일 변화시킬 필요는 없다. 언제나 알맞게 하는 것을 목표로 하자.

만약 신속하고 지지분한 시안 작업에 흥미가 있거나 자신에게 잘 맞는 자체적인 기법이 있다면, 간단히 연락주기 바란다. 우리는 이게 다른 스튜디오에서 얼마나 효과가 있는지 보고 싶다.