게임코드/아트 분류: 공백을 메울 수 있는 테크니컬 아티스트

(The Code/Art Divide: How Technical Artists Bridge The Gap)

제이슨 헤이스(Jason Hayes))
가마수트라 등록일(2008년 8월 20일)

http://www.gamasutra.com/view/feature/1651/the_codeart_divide_how_technical_.php

[이 흥미를 끄는 테크니컬 기사는 본래 Game Developer 매거진에서 출판되었으며, 여기에서 美볼리션(Volition)의 제이슨 헤이스는 세인츠 로(Saints Row) 프랜차이즈 개발자가 테크니컬 아티스트들을 자신의 개발 루트상에 통합시키는 방법에 대해 토론하고 있다.]

예산이 초과되고 팀규모가 지난 세대개발을 통틀어 두 배로 증가하면서 게임개발이 더 복잡해짐에 따라, 아트와 프로그래밍 사이의 틈새를 막기 위한 필요성이 그 어느 때 보다 절실해졌다.

아트 파이프라인을 개발하고 이러한 필요성을 이해하기 위해 프로그래머에게 의존하던 시대는 이미 지나갔다. 오늘날의 게임개발은 훨씬 더 효율적인 작업을 필요로 한다. 즉, 지난 하드웨어 세대에서 보여진 것과 견줄만한 팀규모를 통해 고품질의 트리플 A 급 게임타이틀을 개발할 수 있어야 한다.

테크니컬 아티스트는 자리를 잡기 시작한 게임업계에서 새로운 개념이자 새로운 역할을 맡고 있다. 각 회사마다 테크니컬 아티스트의 역할과 책임감이 무엇인지에 관해 서로 다른 생각을 가지고 있다. 그러나, 개발프로세스를 정말로 극대화시키기 위해서는 기업들이 테크니컬 아티스트가 지닌 역량을 최대한 통합시킬 필요가 있다.

이에 따라 美볼리션(Volition)사는 테크니컬 아티스트들을 게임개발 프로세스 속에 어떻게 통합시킬 지에 관한 청사진을 가지고 있으며, 잘하면, 당신의 게임작업장 역시 이와 동일한 작업을 수행할 수 있는 방법에 관한 아이디어를 제공 받을 수 있다.

실례

Xbox 360 용 세인츠 로(Saints Row) 개발 당시, 나는 아트애셋을 게임 속으로 통합시키는데 사용되는 다수의 툴과 파이프라인을 디자인하고 개발하는 임무를 맡고 있었다. 그러나, 테크니컬 아티스트가 볼리션에 없었을 때는 아티스트들이 사용할 시스템을 개발하고 디자인하는 것은 프로그래머들의 몫이었다.

테크니컬 아티스트를 고용하지 않는 게임작업장과 마찬가지로, 아트 파이프라인을 개발하고 지원해야 할 사람은 대체로 프로그래머들이다. 일반적으로, 아티스트는 요구서를 제출하고, 사용할 툴을 훗날 어떤 시점에서 제시한다. 이들 툴은 대부분 아티스트들의 관점에서 볼 때 사용하기가 쉽지 않거나 워크플로우가 명확하지 않다. 게다가, 툴을 사용하여 문제진단 시 문제점이 나올 수 있는데, 문제처리를 위해 프로그래머가 시간을 내는 것이 일반적으로 어렵고 특히, 프로젝트 사이클의 끝이 다가올수록 더욱 그러하기 때문이다.

테크니컬 아티스트를 작업장 속에 통합시킴으로써 프로그래머들은 게임의 툴과 파이프라인을 개발하고 유지하는 임무를 단독으로 수행하지 않아도 된다. 프로그래머가 툴과 파이프라인의 디자인(및 때로는 실행)에 여전히 관여하고 있는 반면, 테크니컬 아티스트는 이들 뒤에서 추진력이 되어 프로그래머와 아티스트 모두의 이익을 최대한 이끌어 낸다.

이러한 도움으로 인해, 프로그래머는 게임코드 개발에 더 많이 집중할 수 있고 아티스트는 사용하기 쉬운 툴과 워크플로우를 이용하여 가장 보기 좋은 컨텐츠를 만드는데 온 힘을 기울일 수 있다.

절충의 기술

세인츠 로 개발 막바지에 도달했을 때 발생했던 하기의 사례는 테크니컬 아티스트를 활용하는 것이 얼마나 중요한지에 대해 설명해준다.

우리의 게임은 프레임률에 관한 몇 가지 심각한 문제점을 가지고 있었는데, 특히 밤에 게임할 때 문제가 되었다. 여기에는 다수의 이유가 있었지만, 주된 것은 하드웨어를 이용할 수 있기 전에 벌써 게임을 개발했기 때문이었다. 또한, 작업장용으로 사용하기에는 이렇게 개발된 게임이 새로운 유형의 장르였다.

프레임률에 관한 문제를 일으키는 많은 원인 중 하나는 픽셀단위의 역동적인 조명(per-pixel dynamic lighting)을 자유롭게 사용한 것이었다. 시간이 많지 않았기 때문에 많은 프로그래밍측 사람들은 밤에는 역동적인 조명을 끄고 이를 효과로 가장하는 것이 최선이라고 느꼈다. 반면, 아티스트측에서는 조명을 계속 켜두고자 했는데 훨씬 더 나은 신빙성과 풍부함이 조명을 통해 밤의 풍경 속에 제공되기 때문이었다. 모든 것이 동일한 조건하에서, 프로그래머측이 이 싸움에서 이길 수도 있었는데 왜냐하면 안정된 프레임률로 게임을 출시하는 것이 그렇지 않은 경우보다 더 낫기 때문이다.

게임엔진과 이의 역량 및 한계점에 대한 지식이 있었기 때문에, 나는 문제를 해결하기 위해 하이브리드 솔루션을 개발하자고 제안했다. 이것은 게임 플레이어 주변에 일정한 거리에서 역동적인 조명은 그대로 둔 채, 역동적인 조명과 관련된 GPU 및 CPU 비용을 지불하지 않고도 효과를 통해 훨씬 더 잘 조명이 된 도시의 인상을 주는 것이다.

그 밖의 다른 최적화 방안과 더불어 이 솔루션으로 인해 밤시간대 게임플레이 시 프레임률을 극적으로 향상시켰으며 이것이 프로그래머들을 만족시켜 주었다. 시각적으로는, 아티스트들이 원했던 신빙성과 풍부함을 여전히 유지할 수 있었다.

상기에 기술한 것과 같은 경우는 종종 발생하며, 때로는 날마다 벌어질 수 있다. 각각의 부서뿐만 아니라 전반적인 제품을 위해서도 무엇이 최선이고 중요한지에 관한 협상을 위해서는, 이들 두 분야 사이에 경험 있는 테크니컬 아티스트를 갖추는 것이 중요하다.

파이프라인 및 시스템 기획자

일반적으로, 테크니컬 아티스트는 게임에 필요한 모든 아트 파이프라인을 디자인하고 개발할 수 있어야 한다. 이러한 관점에서 볼 때, 테크니컬 아티스트가 지닌 역할 중 하나는 파이프라인 및 시스템 기획자가 되는 것이다. 볼리션에서, 이것은 서로 다른 몇 가지 레벨에서 이루어진다.

시스템에 따라, 우리는 프로그래밍 부서 및 아트 부서와 작업하여 이 둘 부서를 위한 최상의 작업을 결정하고 공통된 견해에 도달하고자 노력한다. 이러한 레벨에서 보면, 우리가 기술지향과 컨텐츠 지향의 규율 사이에서 협상가로서 행동한다고 말할 수 있을 것이다.

그러나, 중요한 게임시스템을 디자인할 때는 Xbox 360 또는 플레이스테이션 3 와 같은 게임엔진과 개발 하드웨어 모두를 잘 알고 있는 테크니컬 아티스트가 요구된다. 필요한 지식의 정도에 관해서는, 만일 테크니컬 아티스트가 충분한 경험이 없거나 신의성실의 의무(due diligence)를 최우선시하지 않으면 파이프라인이 악화된 상황으로 곧바로 바뀔 수 있다. 빈번히, 게임생산의 중간단계나 심지어 개발 사이클 후에도 초창기 실수가 느껴지는 경우가 있다.

테크니컬 아티스트는 이들 시스템을 그 밖의 다른 규율과 조정하여 디자인하고 기술할 뿐만 아니라 변화를 추진하고 옹호하는 사람들이다. 이들은 시스템을 잘 알고 있기 때문에, 파이프라인을 지원하기 위해 작성된 툴내 버그를 없애고 일을 진행시키는 방법 등과 관련해서 정보의 주요 공급원이 된다.

볼리션내 테크니컬 아티스트들이 프로그래머를 위해 게임의 코드구조를 디자인하지 않는다는 사실에 주목해야 한다. 이러한 수준의 정교성은 우리의 작업도 아니며 우리의 전문분야도 아니다. 이와는 대조적으로, 우리는 컨텐츠 생성에서 게임에 이르기까지 요구된 게임피처를 확보하고 이들 사이의 모든 것을 입수하기 위한 최선의 방식을 개발하기 위해 더 높은 레벨에서 프로그래머들과 함께 작업한다.

컨텐츠 예산 및 가이드라인

테크니컬 아티스트는 아트 컨텐츠를 생성하기 위한 테크니컬 가이드라인을 개발하고 예산을 세우는 일 이면에서 추진력을 제공한다. 무엇보다도 테크니컬 아티스트들은 이러한 요건들이 게임성능에 도움을 주는 동안 이들이 균형을 맞추어서 높은 수준의 시각적인 품질을 제공해야 한다는 사실을 기억하고 있어야 한다. 게임의 성능에 방해를 주지 않는 방식으로 아트를 만들어야 하는 것이다. 그보다도, 게임엔진 및 하드웨어 역량이 지닌 이점을 이용하여 이를 생성해야 한다. 이러한 목적을 이루기 위해, 테크니컬 아티스트는 팀의 렌더링 프로그래머들과 긴밀하게 작업한다.

툴 프로그래머

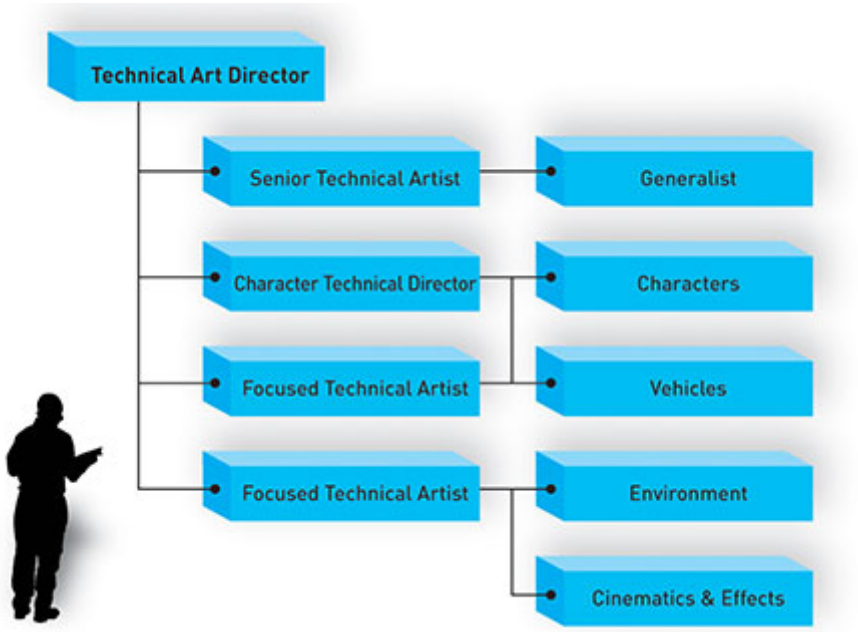
테크니컬 아티스트는 어떠한 도움 없이도 아트 파이프라인용 툴을 개발할 수 있을 정도의 충분한 융통성을 지니고 있어야 한다. 일반적으로, 프로그래머는 이러한 툴을 작성하고자 하지 않으며, 이러한 공백을 채울 수 있는 사람이 바로 테크니컬 아티스트이다.

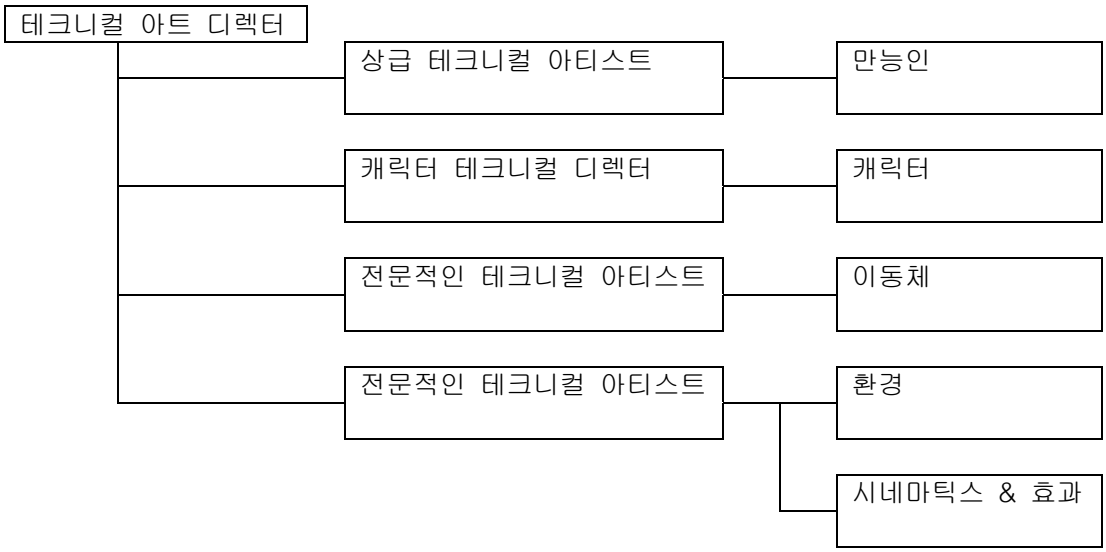
툴은 다양한 레벨로 개발된다. 오늘날의 기준에서 볼 때 대부분의 테크니컬 아티스트들은 아트관련 경험이 있으며 맥스 스크립트(MaxScript)나 멜(Mel)과 같은 역동적인 스크립팅 언어를 즐겨 사용한다.

이들의 흥미와 경험수준이 높아지면서, 일부 테크니컬 아티스트들의 경우 C# 및 C++ 와 같이 훨씬 더 낮은 수준의 언어에 익숙해지고 있다. 이를 통해, 컨텐츠 생성 패키지 외부의 툴 및, 익스포터와 같이 컴파일링된 플러그인을 작성할 수 있는 한층 강화된 능력이 이들에게 주어지고 있다.

전문적인 규율

채용하기 어려운 직위가 있다면, 바로 테크니컬 아티스트일 것이다. 한 때, 테크니컬 아티스트는 필요에 의해 스크립트 작성법을 스스로가 익히고 아트의 기술적인 면에 선천적인 흥미를 가졌던 컨텐츠 아티스트였다. 볼리션에서, 우리는 이러한 역할을 게임개발의 핵심영역으로 분류시키고 있다. 자사의 대표적인 테크니컬 아트팀 구조에 대한 예가 도 1 에 나타나 있다.





도 1: 볼리션의 대표적인 아트팀 구조

테크니컬 아트 디렉터. 테크니컬 아트 디렉터는 우리의 대표적인 팀구조내 아트 디렉터와 동일한 레벨상에 위치한다. 이 사람은 테크니컬 아트팀을 조정하고, 게임피처의 우선사항을 결정하고, 프로젝트 위험성을 식별 및 사정하며, 중요한 툴을 스케줄링 및 디자인하는 임무를 맡고 있다.

또한, 테크니컬 아트 디렉터는 게임시스템 및 파이프라인을 디자인 및 이행하며, 아트 컨텐츠 생성용 예산과 가이드라인을 세우고, 게임의 렌더링 성능이 최적으로 수행되는지 여부를 확인하면서, 높은 수준의 시각적인 품질을 유지하기 위해 아트 디렉터와 함께 작업한다.

만능인(Generalist). 만능인이라 부르는 사람은 일반적으로 게임내 어떠한 시스템도 운영할 수 있는 상급 테크니컬 아티스트를 가리킨다. 이들은 광범위한 경험을 갖추고 있으며 일반적으로 중요한 사람들이다.

캐릭터 테크니컬 디렉터. 캐릭터 테크니컬 디렉터는 캐릭터의 뼈대를 세우고, 캐릭터를 차려입히며, 모션캡처와 애니메이션의 필수품을 식별 및 사정하고, 애니메이터를 스케줄링 및 조정하며, 툴과 파이프라인을 개발 또는 디자인함으로써 게임의 캐릭터 시스템을 지원한다.

상급 테크니컬 아티스트. 상급 테크니컬 아티스트는 규모가 더 크고 더 중요한 게임시스템 및 파이프라인을 디자인하고 이행하는 임무를 주로 맡고 있다. 또한, 이 사람은 렌더링 성능뿐만 아니라 높은 시각적인 품질에 대해서도 최적의 방식으로 컨텐츠가 생성되는지 여부를 보장할 책임 역시 일부 가지고 있다.

전문적인 테크니컬 아티스트. 전문적인 테크니컬 아티스트는 일반적으로 초급 내지 중급단계의 테크니컬 아티스트를 일컫는다. 이들은 환경아트나 캐릭터 아트와 같이 게임의 특정영역에 주로 포커스를 맞춘다. 이들 전문적인 테크니컬 아티스트들은 자신들의 특정한 아트부서에서 선두적인 지위를 차지하며 만능인이나 테크니컬 아트 디렉터측을 통해 승인을 얻는다. 또한, 이들은 직접적인 지원을 하고, 각각의 자신들의 부서에 필요한 파이프라인 및 필수적인 툴을 개발한다.

기술적인 지원

테크니컬 아티스트 소관인 또 다른 주요 영역으로는 아트팀에 기술적인 지원을 해주는 것이 있다. 여기에는 아티스트가 컨텐츠 생성 패키지(3ds Max, Maya) 또는 그 밖의 다른 사내(社內) 독점적인 툴에 관한 문제를 진단하는 작업과 같은 임무가 포함된다.

아티스트의 지원요구에 응답하는 것은 시간이 소모되는 프로세스이다. 테크니컬 아티스트를 위한 이러한 프로세스를 빠르게 진행시키기 위해, 스크린샷을 첨부하여 에러나 문제를 기술적으로 설명해주는 이메일을 통해 모든 요구사항이 전달되어야 한다.

우리가 이렇게 하는 데는 몇 가지 이유가 있다. 볼리션의 테크니컬 아티스트들은 아웃소싱된 사람들까지 포함하여 방대한 수의 아티스트를 지원한다. 빠른 응답을 통해 요구사항을 해결하는데 걸리는 시간을 줄이고 아티스트를 보충하여 이들이 가능한 방해를 덜 받으면서 지속적으로 작업할 수 있도록 하는 것이 중요하다.

스케줄링

테크니컬 아트팀이 하는 많은 일들(팀이 개발프로세스를 얼마나 진행했는지 여부)이 이들 자신이나 프로젝트의 스케줄링에 영향을 끼친다. 개별적으로, 테크니컬 아티스트들은 종종 순간적인 통보를 받으면서 개발툴을 진행하고 당면과제를 해결해야 하는 급한 상황 사이를 자주 왔다 갔다 한다. 다중규율을 받은 팀구성원으로써, 이들은 다른 많은 부서의 미팅에도 참석요구를 받으며, 이 중 예기치 않은 미팅도 잡혀있을 때가 있다.

규율을 스케줄링하기가 어렵다는 사실을 관리자가 받아들이고 개방된 자세를 유지한다면, 스케줄링 테크니컬 아티스트는 통제하기 쉽다.

테크니컬 아티스트의 스케줄을 맞출 때 유익하게 이용되는 것으로 나타난 몇 가지 접근방법이 하기에 나와 있다.

요구사항을 초기에 식별한다. 볼리션내 작업장 사람들은 모두 이메일 가명을 통해 피처나 새로운 툴에 대한 아이디어를 프로젝트에 제출하도록 한다. 그러면, 모든 프로젝트 지도부가 이들 아이템을 검토하고 이 단계에서 종속 사항들을 식별한다.

요구사항을 평가할 때 우리는 5 점으로 분류된 점수를 사용하는데, 1 은 "필수사항"을 의미한다. 우리는 1 과 2 로 표시된 모든 요구사항에 우선권을 부여하며, 3 내지 5 로 책정된 아이템들의 경우 스케줄내 빈 곳을 채우기 위해 나중에 재검토한다.

신의성실의 의무(Due diligence). 모든 툴이나 피처에 대한 요구사항은 다음과 같은 3 단계의 프로세스를 거친다: 조사, 이행, 그리고 문서조사.

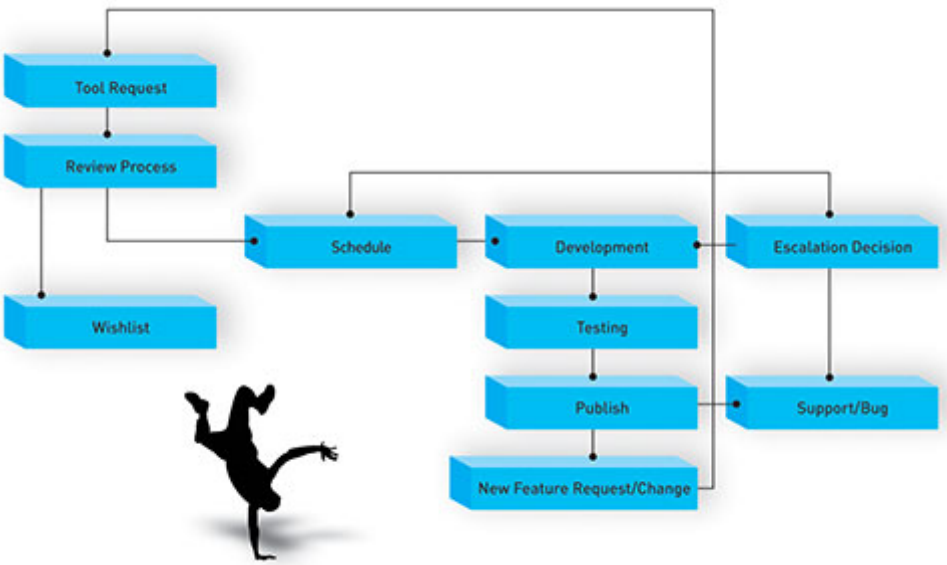
요구사항의 위험성을 식별하기 위해 조사단계를 거친다. 너무 자주, 스케줄내에 위험성이 설명되지 않고 있다: 이를 식별함으로써 견고한 툴을 구축하는 것이 중요하다는 사실을 명심하게 한다.

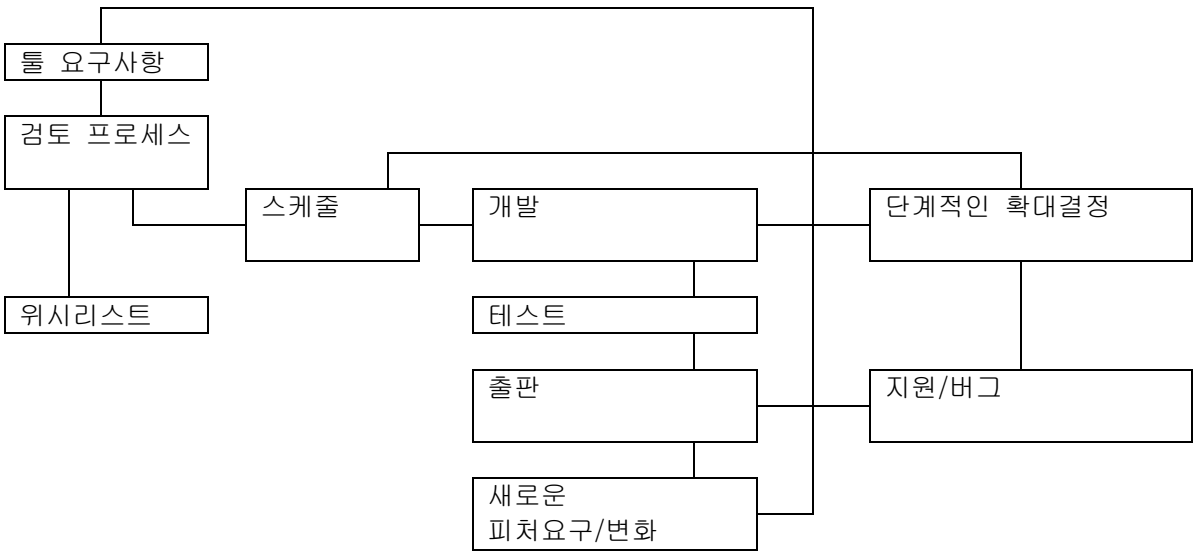
종종 문서조사를 별 것 아닌 것으로 치부해 버린다. 양호한 문서조사를 통해, 툴을 사용하는 사람들 모두가 이들이 지닌 기술적인 능력에 관계없이 툴을 제대로 사용할 수 있도록 보장할 수 있다.

스케줄 지원시간. 이것은 갑자기 변하는 상황을 받아들이는 것이다. 우리의 경험상, 이정표나 최종기한의 끝에 가까이 다가갈수록 지원시간이 일반적으로 빠르게 증가하였다

스케줄 완충시간. 스케줄링된 지원시간에도 불구하고, 예견할 수 없는 일들이 불가피하게 생긴다.

변화관리. 스케줄을 맞추는데 있어 툴 및 피처에 대한 요구사항에 관해 견고한 변화관리 계획을 이행하는 것이 필수적이다. 이행하기에 중요하지 않은 것처럼 보이는 기존의 툴에 대한 피처가 너무도 자주 요구되지만, 이들 모두를 다 들어주다 보면 이미 충분히 복잡한 스케줄링 기술아트 문제가 더 복잡해질 뿐이다. 도 2 는 우리가 피처 요구사항을 이행한 방법과 우리의 변화관리 계획을 도시하는 플로우차트이다.





도 2: 볼리션의 피처 요구사항 및 변화관리 계획에 대한 플로우차트.

이행

한 기업내에 얼마나 많은 테크니컬 아티스트가 있어야 할까? 대략 80 내지 90 명의 사람들이 있는 볼리션의 팀규모에서 볼 때, 지난 몇 년 동안 우리는 프로젝트당 3 명 또는 4 명의 테크니컬 아티스트가 필요하다는 사실을 발견했다.

우리가 구성한 팀은 테크니컬 아트 디렉터인 선두와, 적어도 한 명의 상급 테크니컬 아티스트, 그리고 한 명의 캐릭터 테크니컬 디렉터로 구성된다. 그 밖의 다른 팀들은 게임의 일정 영역에 특별히 전념하도록 배정된 테크니컬 아티스트에 더 많이 집중하고 있다.

이러한 역할을 수행할 적임자를 찾는 것은 어려운 일이지만, 오늘날의 경쟁적이고 고비용이 드는 환경 속에서 이를 가볍게 지나쳐서는 안 된다. 당신의 작업장에 테크니컬 아티스트가 한 명도 없거나, 잠재력을 완벽하게 사용하지 못한 이들이 일부 존재한다면, 이들을 다시 한번 살펴보길 바란다. 그러면, 만족할만한 결과를 얻을 것이다.