

## 제23장

## 아키텍처 다이어그램 작성

신입 소프트웨어 아키텍트는 처음 이 직책을 맡게 된 계기가 된 기술적 지식과 경험 외에도 업무가 얼마나 다양한지 알고 놀라는 경우가 많습니다. 특히 효과적인 의사소통은 아키텍트의 성공에 매우 중요합니다. 아무리 훌륭한 기술적 아이디어를 가지고 있더라도, 관리자들이 그 아이디어에 투자하고 개발자들이 구현하도록 설득하지 못한다면, 그 뛰어난 능력은 결코 빛을 발할 수 없습니다.

다이어그램 작성은 건축가에게 필수적인 의사소통 기술입니다. 각 주제에 대해 다른 책들이 수없이 많지만, 여기서는 각 주제별 핵심 사항 몇 가지를 짚어보겠습니다.

건축물을 시각적으로 묘사하기 위해 건축가는 종종 다양한 관점을 제시해야 합니다. 예를 들어, 전체적인 구조를 개괄적으로 보여준 다음 건축물의 특정 부분에 대한 설계 세부 사항으로 점차 자세히 살펴볼 수 있습니다. 그러나 전체 건축물 내에서의 위치를 명시하지 않고 특정 부분만 보여 주면 보는 사람을 혼란스럽게 할 수 있습니다.

표현 일관성이란 아키텍처의 각 부분 간의 관계를 시점을 변경하기 전에 항상 보여주는 관행이며, 다이어그램과 프레젠테이션 모두에서 똑같이 중요합니다. 예를 들어, 실리콘 샌드위치 솔루션에서 플러그인들이 서로 어떻게 연관되는지 자세히 설명하려면, 먼저 시스템의 전체 토폴로지를 보여주는 아키텍처 다이어그램을 제시하고, 그 토폴로지와 플러그인 구조 간의 관계를 보여준 다음, 플러그인 구조 자체를 설명하는 순서로 진행해야 합니다. **그림 23-1**은 이러한 예시를 보여줍니다.

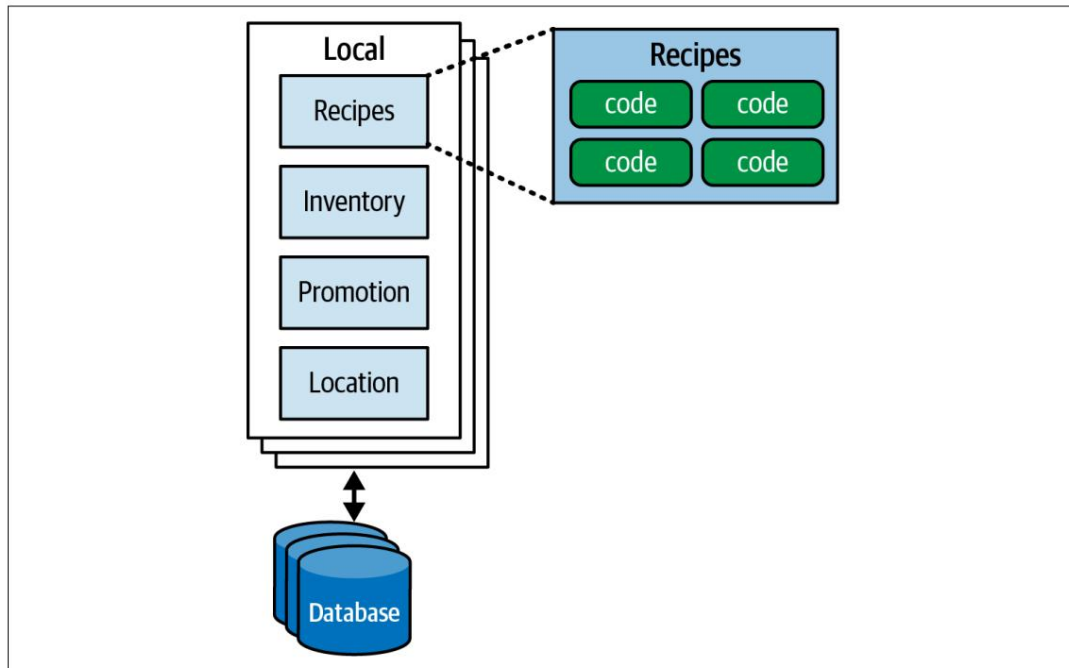


그림 23-1. 더 큰 다이어그램에서 맥락을 나타내기 위해 표현의 일관성을 사용하는 방법

시청자가 제시된 항목의 범위를 이해하고 흔히 발생하는 혼란의 원인을 제거할 수 있도록 표현의 일관성을 신중하게 활용하십시오.

## 도표 작성

아키텍처의 토폴로지는 구조가 어떻게 서로 연결되는지를 보여주고 팀 간의 귀중한 공통된 이해를 형성할 수 있도록 해주기 때문에 건축가와 개발자 모두에게 항상 중요한 관심사입니다. 모든 건축가들이 다이어그램 작성 능력을 최대한으로 연마하도록 권장합니다.

## 도구

현재 건축가를 위한 다이어그램 도구는 매우 강력하며, 모든 건축가는 자신이 선택한 도구를 깊이 있게 익혀야 합니다. 하지만 특히 설계 초기 단계에서는 저해상도 설계 산출물을 소홀히해서는 안 됩니다. 초기에 매우 일시적인 설계 산출물을 만들어 두면 건축가가 자신이 만든 결과물에 지나치게 집착하는 것을 방지할 수 있는데, 이를 우리는 '비합리적인 산출물 집착'이라고 부릅니다.

## 비합리적인 인공물 애착

비합리적 결과물 애착 안티패턴은 어떤 결과물에 대한 개인의 비합리적인 애착과 그 결과물을 만드는 데 걸린 시간 사이의 비례 관계를 설명합니다. 예를 들어, 건축가가 Visio와 같은 도구를 사용하여 아름다운 다이어그램을 만드는 데 4시간을 투자했다면, 2시간만 투자했을 때보다 그 결과물에 훨씬 더 비합리적으로 애착을 가질 것입니다.

애자일 소프트웨어 개발 방식의 장점 중 하나는 최소한의 형식과 절차로 필요한 산출물을 즉시만 들어낼 수 있다는 점입니다. 많은 애자일 개발자들이 인덱스 카드와 포스트잇을 선호하는 이유도 바로 여기에 있습니다. 간단한 도구를 사용하면 잘못된 부분을 과감히 버릴 수 있어 자유롭게 실험하고, 수정, 협업, 토론을 통해 산출물의 진정한 본질을 발견할 수 있기 때문입니다.

대표적인 일시적인 기록물은 화이트보드에 그려진 그림을 휴대전화로 찍은 사진입니다(물론 "지우지 마시오!"라는 문구는 필수죠). 요즘 많은 건축가들은 화이트보드 대신 오버헤드 프로젝터에 연결된 태블릿을 사용하고 있습니다.

이러한 방식은 여러 가지 장점을 제공합니다. 첫째, 태블릿은 무제한의 작업 공간을 제공하여 팀에서 필요한 만큼의 그림을 그릴 수 있습니다. 둘째, 팀원들이 "만약 ~라면" 시나리오를 복사/붙여넣기하여 활용할 수 있는데, 이는 화이트보드에 직접 그릴 경우 원본을 가리는 것을 방지합니다. 셋째, 태블릿에서 생성된 이미지는 이미 디지털화되어 있어 화이트보드 사진에서 흔히 발생하는 빛 반사가 없습니다. 넷째, 전자 이미지를 사용하면 원격 근무가 훨씬 수월해지고 협업이 활발해집니다.

결국에는 멋진 도구를 사용하여 보기 좋은 다이어그램을 만들어야 할 것입니다. 하지만 그 시간을 투자하기 전에 팀이 디자인을 충분히 검토했는지 확인하세요. 어떤 플랫폼을 사용하든 강력한 다이어그램 도구를 찾을 수 있을 것입니다. 저희는 이 책에 나오는 모든 다이어그램의 초기 버전을 **OmniGra** **le**를 사용하여 만들었고, 이후 O'Reilly 일러스트레이터들이 다듬었습니다. 특정 도구를 추천하는 것은 아니지만, 기본적으로 다음과 같은 기능을 갖춘 도구를 선택하는 것이 좋습니다.

### 레이어 드

로잉 도구는 종종 레이어 기능을 지원하며, 건축가는 이러한 레이어 기능을 잘 활용하는 방법을 익혀야 합니다.

레이어 기능을 사용하면 항목들을 논리적으로 그룹화하여 필요에 따라 표시하거나 숨길 수 있습니다. 예를 들어, 프레젠테이션을 위해 포괄적인 다이어그램을 만들되, 직접적으로 다루지 않을 때는 복잡한 세부 정보를 숨길 수 있습니다. 또한 레이어는 단계적으로 구성되는 그림을 표현하는 데에도 유용합니다.

### 스텐실/템플릿 스텐

실 도구를 사용하면 다른 기본 도형의 조합을 포함하여 일반적인 시각적 구성 요소 라이브러리를 구축할 수 있습니다. 예를 들어, 이 책 전체에서 개별 마이크로서비스와 같은 항목에 대한 표준 아이콘을 보셨을 텐데, 이러한 아이콘은 저자들이 만든 스텐실 도구에 단일 항목으로 존재합니다. 조직 내에서 공통적으로 사용되는 패턴과 아티팩트에 대한 스텐실 세트를 구축하면 아키텍처 다이어그램 전반에 걸쳐 일관성을 유지하고 새로운 다이어그램을 더 빠르게 작성할 수 있습니다.

### 자석 기능

은 많은 그리기 도구에서 도형 간의 선을 그릴 때 도움을 주는 기능입니다. 자석은 선이 자동으로 연결되어 정렬 및 기타 시각적 효과를 향상시키는 도형 위의 위치를 나타냅니다. 일부 도구는 사용자가 자석을 추가하거나 직접 만들 수도 있습니다.

레이어를 장식적인 용도가 아닌 의미론적인 용도로 사용하세요. 레이어 기능

을 지원하는 드로잉 도구를 선호하지만, 레이어를 전체적인 이미지에 의미를 부여하는 방식으로 사용하는 것을 권장합니다. 예를 들어, 각 다이어그램의 기본 레이어는 아키텍처의 토폴로지를 나타내야 합니다. 컨테이너, 데이터베이스, 종속성, 브로커 및 기타 핵심 요소들을 포함해야 합니다. 이 레이어는 구현보다는 아키텍처에 초점을 맞춰야 합니다. 예를 들어, 특정 프로토콜을 명시하는 대신 "동기 통신"과 같은 표현을 사용하는 것이 좋습니다. 다음 레이어에는 구현 세부 정보(데이터베이스 유형, 통신 프로토콜 등)를 포함해야 합니다.

이러한 방식으로 레이어를 사용하면 다이어그램을 확장할 수 있습니다. 도메인 중심 설계 경계, 트랜잭션 범위 또는 아키텍트가 아키텍처 토폴로지와 대조하여 표시하고자 하는 기타 메타 정보를 나타내기 위해 다른 텍스트화된 레이어를 추가할 수 있습니다.

이러한 구체적인 유용한 기능 외에도, 해당 도구는 당연히 선, 색상, 도형 및 기타 시각적 요소와 같은 기본 기능을 지원해야 하며, 다양한 형식으로 내보낼 수 있어야 합니다.

다이어그램 표준: UML, C4 및 ArchiMate 소프트웨어 업계에서는

기술 다이어그램을 작성하기 위해 여러 가지 공식 표준을 사용합니다. 여기서는 가장 널리 사용되는 세 가지 표준을 살펴보겠습니다.

## UML

그래디 부치, 아이바르 제이콥슨, 그리고 짐 램바우는 1980년대에 서로 경쟁하던 설계 철학을 통합하기 위해 UML(Unified Modeling Language) 표준을 만들었습니다.

모든 장점을 갖춘 최상의 방안으로 여겨졌지만, 위원회에서 설계된 많은 것들과 마찬가지로 사용을 의무화한 조직 외에는 큰 영향력을 발휘하지 못했습니다.

건축가와 개발자는 여전히 UML 클래스 다이어그램과 시퀀스 다이어그램을 사용하여 구조와 워크플로를 전달하지만, 다른 대부분의 UML 다이어그램 유형은 더 이상 사용되지 않습니다.

## C4

C4는 2006년부터 2011년 사이에 사이먼 브라운이 UML의 단점을 보완하고 접근 방식을 현대화하기 위해 개발한 다이어그램 기법입니다. C4의 "네 가지 C"는 다음과 같습니다.

### 문맥

사용자 역할 및 외부 종속성을 포함한 시스템의 전체 맥락.

### 컨테이너

아키텍처 내의 물리적(그리고 종종 논리적) 배포 경계 및 컨테이너. 이 관점은 운영 팀과 아키텍트가 만나기에 좋은 기준점이 됩니다.

### 구성 요소 관점

은 시스템을 구성 요소별로 바라보는 관점으로, 아키텍트의 시스템 관점과 가장 잘 부합합니다.

## C4 클

래스는 UML과 동일한 스타일의 클래스 다이어그램을 사용하는데, 이는 효과적이므로 교체할 필요가 없습니다.

C4는 다이어그램 작성 기법을 표준화하려는 모든 회사에 훌륭한 대안을 제공합니다. 개발자들은 오랜 기간 동안 활발하게 활동하며 많은 지지자를 확보해 왔습니다. 특히, 소프트웨어 개발 생태계의 변화에 맞춰 새로운 기능을 활용하며 지속적으로 확장해 왔습니다. 많은 다이어그램 작성 도구에 C4 다이어그램 템플릿이 포함되어 있으며, **C4 생태계**는 아키텍트를 지원하는 다양한 도구와 프레임워크를 제공합니다. C4는 구성 요소, 선, 컨테이너, 데이터베이스 및 기타 일반적인 아티팩트에 대한 표준을 정의합니다.

실리콘 샌드위치 모듈형 모놀리스 데이터 설계의 C4 다이어그램은 **그림 23-2에 나와 있습니다.**

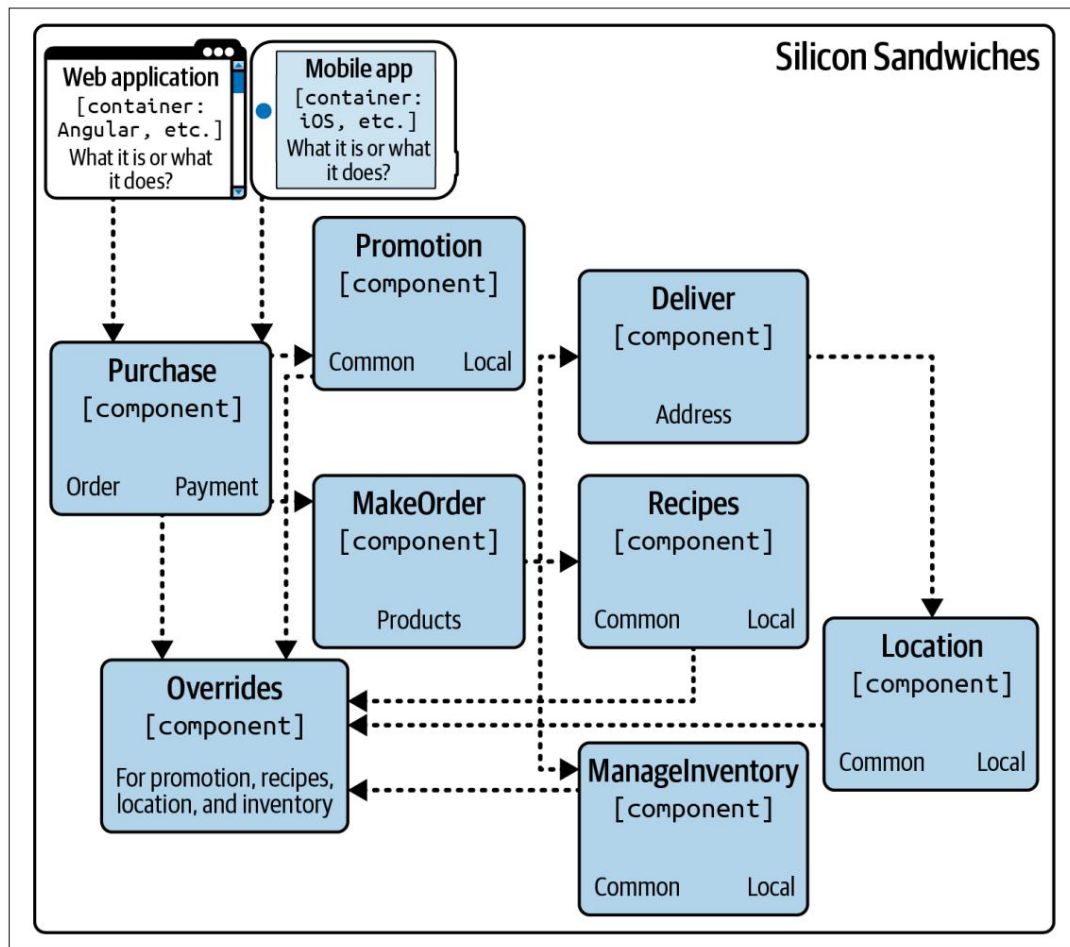


그림 23-2. C4의 실리콘 샌드위치 구성 요소 다이어그램

#### 아키메이트

ArchiMate(architecture와 animate의 합성어)는 The Open Group에서 개발한 오픈 소스 엔터프라이즈 아키텍처 모델링 언어로, 비즈니스 도메인 내외부의 아키텍처를 설명, 분석 및 시각화하는 데 사용됩니다. ArchiMate는 엔터프라이즈 생태계를 위한 보다 간소화된 모델링 언어를 제공합니다. 이 기술 표준의 목표는 모든 예외 상황을 포괄하는 것이 아니라 "가능한 한 간결하게" 만드는 것입니다. 아키텍트들 사이에서 널리 사용되는 언어입니다.

#### 다이어그램 작성 지침 모든

설계자는 자신만의 다이어그램 스타일을 구축해야 합니다. 자체 모델링 언어를 사용하든, 공식적인 언어 중 하나를 사용하든 상관없습니다. 특히 효과적이라고 생각하는 표현 방식을 차용하는 것도 괜찮습니다. 이 섹션에서는 기술 다이어그램 작성에 대한 몇 가지 일반적인 지침을 제공합니다.

## 제목

다이어그램의 모든 요소에는 제목이 있어야 합니다. 단, 청중에게 매우 잘 알려진 요소는 예외입니다. 회전 및 기타 효과를 사용하여 제목이 적절한 요소에 표시되도록 하고 공간을 효율적으로 활용하세요.

## 윤곽

선은 명확하게 보이도록 충분히 굵어야 합니다. 선이 정보의 흐름을 나타내는 경우, 방향 또는 양방향 통행을 나타내기 위해 화살표를 사용하십시오. 화살표 머리의 모양에 따라 다른 의미를 전달할 수 있지만, 일관성을 유지해야 합니다.

아키텍처 다이어그램에서 몇 안 되는 일반적인 기준 중 하나는 실선은 거의 항상 동기식 통신을 나타내고 점선은 비동기식 통신을 나타낸다는 것입니다.

## 형태 앞

서 설명한 형식 모델링 언어들은 모두 표준 형태를 가지고 있지만, 소프트웨어 개발 분야 전반에 걸쳐 통용되는 표준 형태는 없습니다. 대부분의 아키텍트는 자신만의 표준 형태를 만들고, 때로는 이러한 형태들이 조직 전체에 채택되어 표준 언어를 형성하기도 합니다.

우리는 일반적으로 배포 가능한 아티팩트를 나타낼 때는 3차원 상자를, 컨테이너를 나타낼 때는 직사각형을, 데이터베이스를 나타낼 때는 원통을 사용하지만, 그 외에 특별한 기준은 없습니다.

## 라벨

도표의 모든 항목에는 라벨을 붙이는 것이 중요하며, 특히 모호할 가능성이 있는 경우에는 더욱 그렇습니다.

## 색상

건축가들은 종종 색상을 충분히 활용하지 않습니다. 오랫동안 책은 필요에 따라 흑백으로 인쇄되었기 때문에 건축가와 개발자들은 단색 도면에 익숙해졌습니다. 저희는 여전히 단색을 선호하지만, 각각의 결과물을 구별하는 데 도움이 될 때는 색상을 사용합니다. 예를 들어, 19장에서 마이크로서비스 아키텍처 양자역학을 논할 때 그림 19-6 (여기서는 그림 23-3으로 재현) 에서 회색 음영을 사용하여 그룹을 나타냅니다.

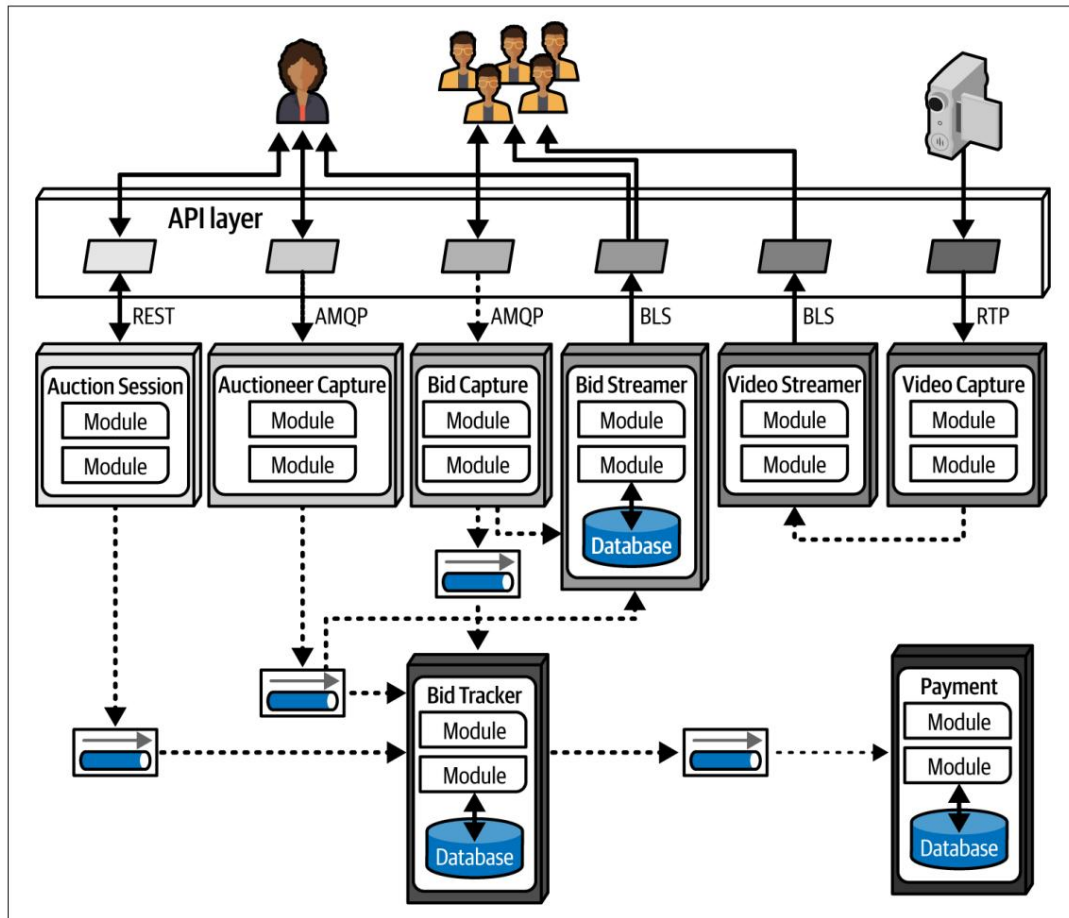


그림 23-3. 19장의 마이크로서비스 통신 예시를 재현한 그림으로, 다양한 서비스를 화색 음영으로 나타낸다.

하지만 중요한 차이점을 나타내기 위해 색상을 사용할 때는 주의해야 합니다. 색맹이거나 시각 장애가 있는 사람들은 그 차이를 구분할 수 없기 때문입니다. 따라서 색상을 구분하지 못하는 사람도 의미를 이해할 수 있도록 색상 외에도 고유한 아이콘을 사용하는 것이 좋습니다. 예를 들어, 횡단보도 신호등은 초록색은 진행, 빨간색은 정지를 나타내지만 각 색상에 맞는 고유한 그림을 표시합니다.

#### 범례

만약 도표의 도형이 어떤 이유로든 모호하다면, 각 도형이 무엇을 나타내는지 명확하게 표시하는 범례를 포함하십시오. 쉽게 오해될 수 있는 도표는 아예 도표가 없는 것보다 더 나쁩니다.



## 요약

다이어그램 표준은 조직에서 일관된 의사소통을 제공하는 데 유용한 방법입니다. 그러나 건축가들은 특히 표준이 설계를 표현하는 좋은 방법을 제시하지 못할 때 규칙을 어기는 경우가 많습니다. 우리는 조직이 표준을 수립하되 합리적인 예외를 허용할 것을 권장합니다.

과거에는 고사양 컴퓨터 지원 소프트웨어 엔지니어링(CASE) 도구가 일반적이었기 때문에 설계자들은 간단한 것들을 표현하기 위해 정교한 모델을 구축해야 했습니다.

그들은 특정 맥락에서 단순히 불필요한 잡음에 불과한 많은 세부 사항을 포함해야 하는 경우가 많았습니다. 우리는 특히 설계 과정 초기에 간편한 다이어그램 도구와 빠르고 간편하게 만들 수 있는 결과물을 선호합니다. 다만, 자신이 만든 결과물에 너무 심취하여 객관성을 잃지 않도록 주의해야 합니다.

