



# SW 엔지니어와 기획자를 위한 프리젠테이션



송태국(tsong@nextree.co.kr)

# 목차 (Table of Contents)

---

1. 소통의 중요성
2. 표현 원칙
3. 실습 #1 – 다른과 같음
4. 기본 그리기
5. 실습 #2 – 레이아웃
6. 실습 #3 – 도표
7. 실습 #4 – 텍스트 표현
8. 실습 #5 – 다이어그램과 그래프
9. 유용한 팁
10. 템플릿
11. 문장
12. 전달(presentation)

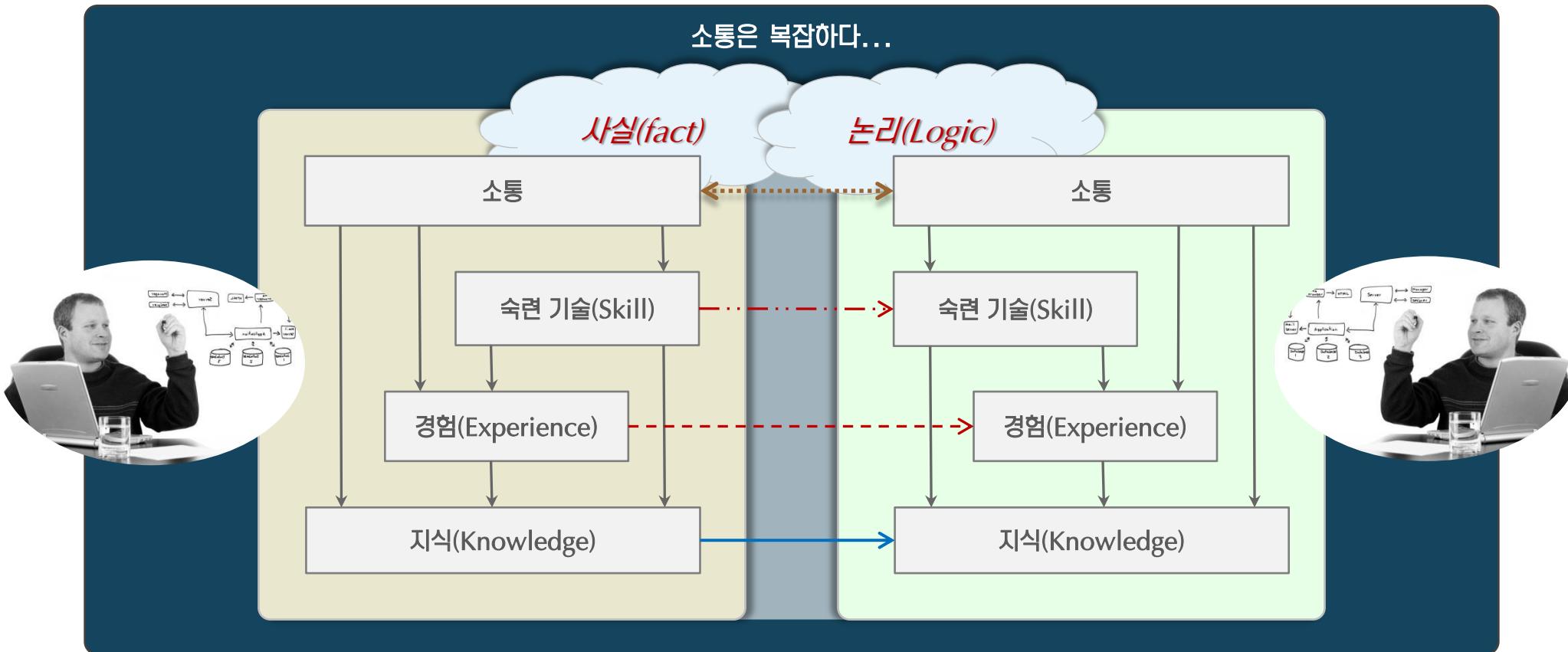


## 1. 소통(communication)의 중요성

- 
- 1.1 소통의 복잡성
  - 1.2 소통은 표현과 발표
  - 1.3 소통의 중요한 역할들
  - 1.4 아키텍트 역량
  - 1.5 소통과 작업
  - 1.6 요약

# 1.1 소통의 복잡성

- ✓ 지식 전달은 가장 단순한 소통으로 상대적으로 명확합니다. 전달 효과 또한 바로 나타납니다.
- ✓ 경험 전달은 “간접적”일 수 밖에 없으며, 전달 효과는 다양하며, 받아들이는 사람의 경험에 따라 폭과 깊이가 결정됩니다.
- ✓ 숙련 기술은 결국 숙련되는데 필요한 시간이 필요합니다. 따라서 기술이란 시간을 통해서 서서히 전수되는 것입니다.
- ✓ 소통은 서로 다른 지식, 경험, 기술을 가진 사람들이 서로의 지식, 경험, 기술을 공유하는 과정으로, 쉽지 않은 과정입니다.



## 1.2 소통은 표현과 발표

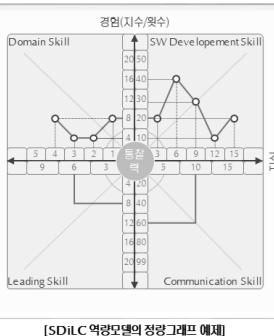
- ✓ 모든 일이 그러하듯이, 소통을 위해 얼마나 고민하고 준비를 하는가에서 승부가 갈립니다.
  - ✓ 충분한 고민을 통해서 소통하고자 하는 내용의 본질을 파악하고, 그 내용을 잘 표현(representation) 합니다.
  - ✓ 잘 표현된 자료를 가지고 있다면, 그리고 그 자료를 만드는 일에 충실했다면 발표의 절반은 끝난 것과 같습니다.
  - ✓ 발표에 필요한 기술은 생각보다 작습니다. 자료를 만드는 과정이 몇 배는 더 중요합니다.

## 1.1 다각형 표현 (2/2)

- ✓ 눈금 및 안내선 창에서 눈금 설정 – 간격은 팀에서 통일해야 합니다. 0.2cm를 추천합니다.
  - ✓ 객체 하나만 제대로 그리면, 나머지는 복사하여 복사하여 붙이기 수준으로 처음 하나를 그릴 때 정성을 다합니다.
  - ✓ 세밀한 조정 (Control + 화살표 키)은 가급적 하지 않습니다. 복사하여 다른 곳으로 이동할 때 그림이 깨어집니다.
  - ✓ 사각형은 정확히 맞출 수 있지만 원은 맞추기 힘듭니다. 세밀한 조정을 하지 않는 선에서 도형을 마무리 짓습니다.



### [SDLC 역량모델의 정량그래프]



### [SDiLC 역량모델의 정량그래프 예제]

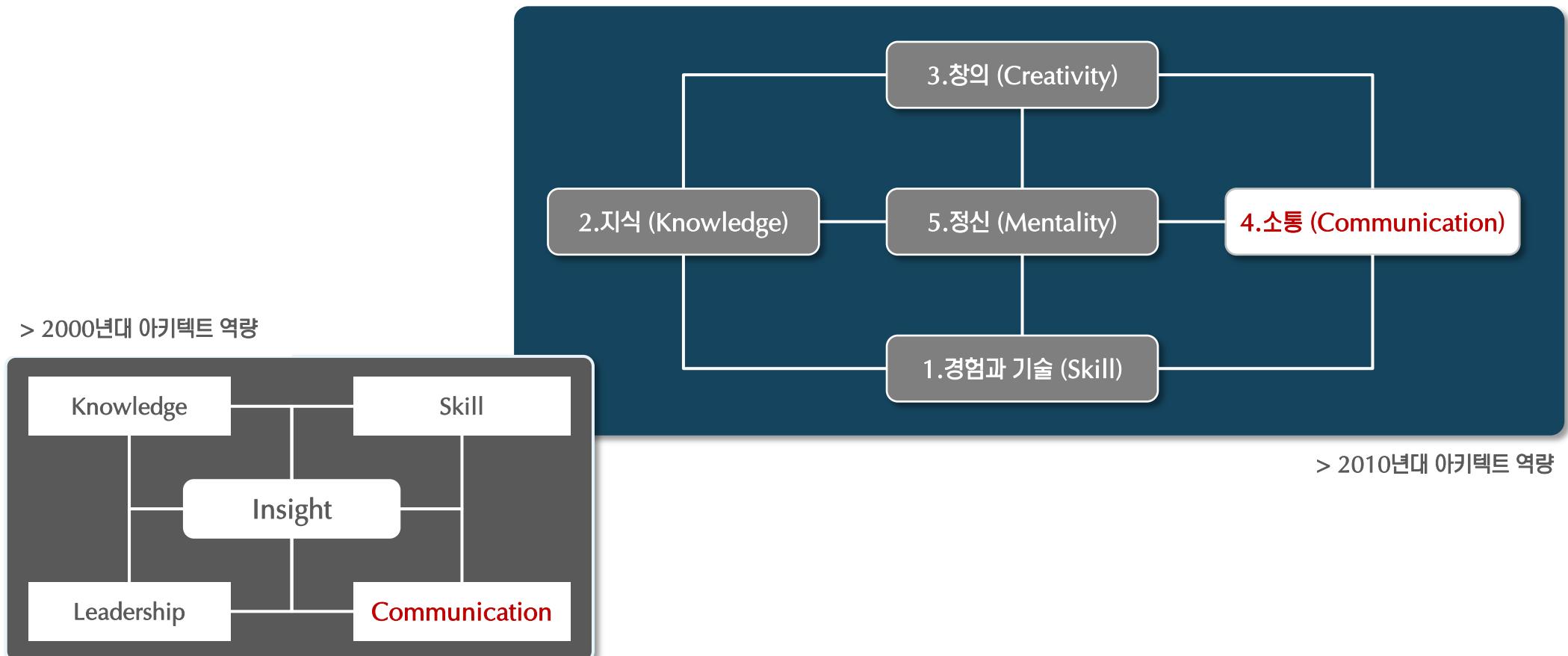


### [SDiLC 역량모델의 정량그래프]



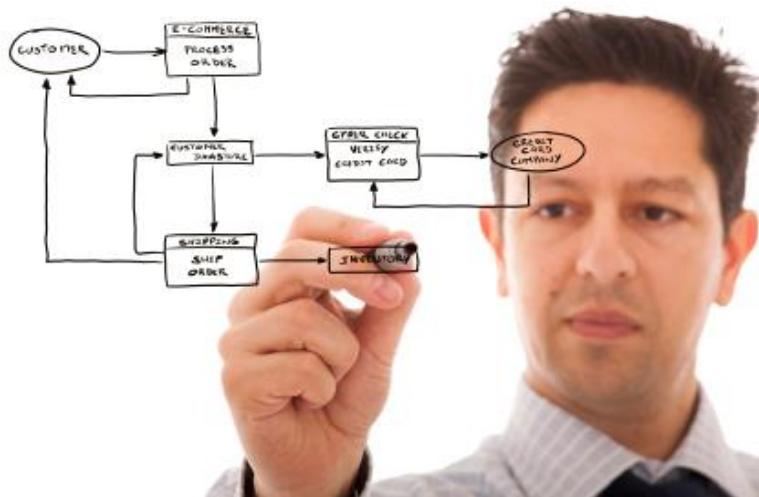
## 1.3 소통의 중요한 역할들

- ✓ SW 엔지니어링에서 위쪽으로 올라갈 수록 소통 역량이 중요합니다. PM, 도메인 모델러, SW 아키텍트
- ✓ 작은 팀이라도 팀 리더가 된다면, 이미 소통의 문제에 부딪힙니다. 소통의 문제는 팀을 쉽게 좌초시켜 버립니다.
- ✓ SW 아키텍트는 최고의 엔지니어입니다. 소통 경로가 가장 복잡한 역할 담당자입니다.
- ✓ SW 아키텍트가 갖추어야 할 역량이 시대의 흐름에 따라 변하지만, 소통(communication) 역량은 변함이 없습니다.



## 1.4 아키텍트 역량 (1/5) – 경험과 기술

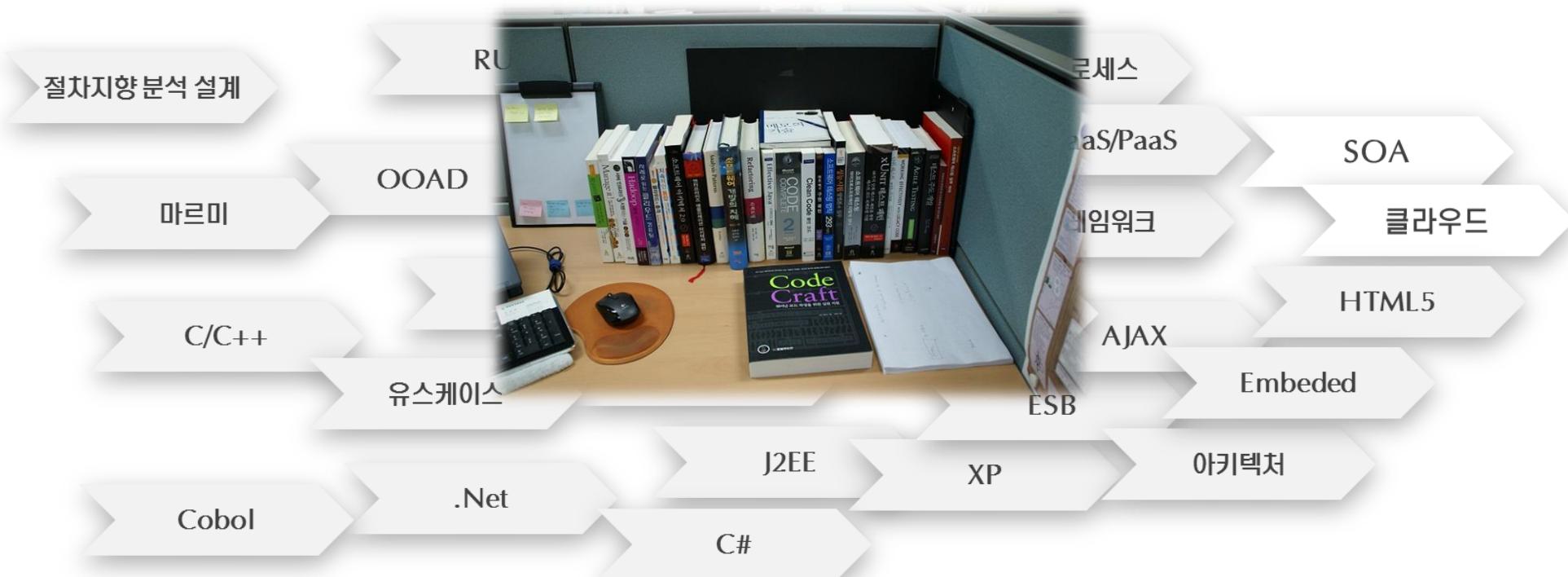
- ✓ 다양한 시스템 구축 경험을 통해 얻는 숙련 기술(skill)은 SW 아키텍트의 첫 번째 자산입니다.
- ✓ 모든 경험은 선(善)이며, 아키텍트는 경험으로부터 깨닫고 배워야 합니다. 충분한 프로젝트 경험이 필요합니다.
- ✓ 경험을 통해서 기술이 성장하며, 기술의 성장을 통해서 보다 높은 수준의 경험을 이끌어 낼 수 있습니다.
- ✓ 경험의 다양성은 경험 상호 간에 시너지를 발휘하는 기반이 되고, 더 나아가 창의적인 설계를 끌어내는 기반이 됩니다.



[출처] <http://www.userzoom.com/news-feed/top-5-user-experience-tips-generate-online-success>

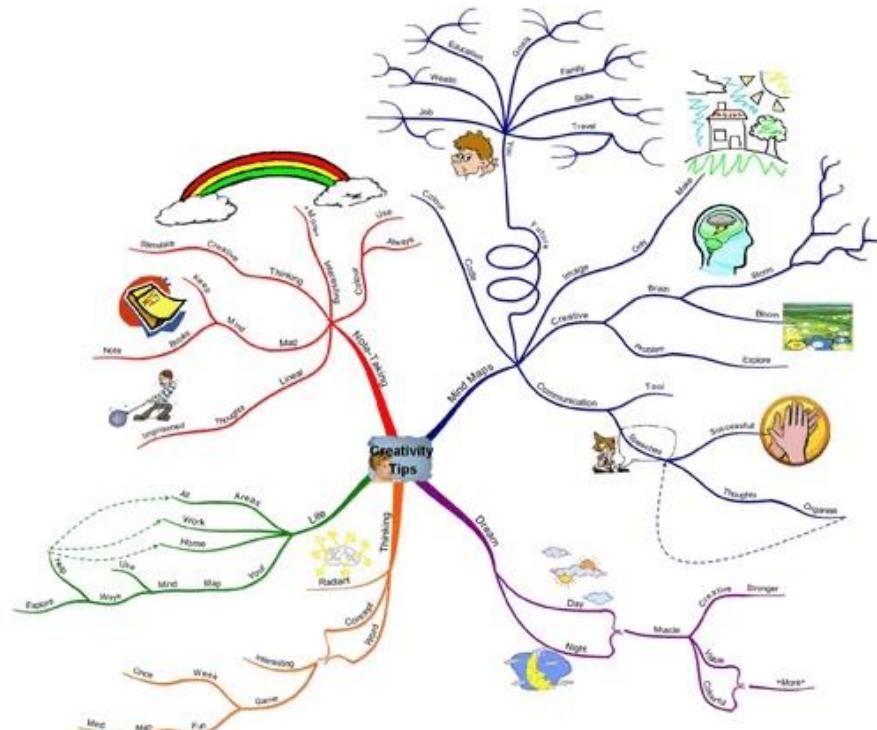
## 1.4 아키텍트 역량 (2/5) – 지식 (Knowledge)

- ✓ SW 엔지니어는 새로운 기술의 홍수 속에서 일합니다. 한 기술을 익히고 나면, 그 기술은 곧 예전 기술이 되어 버립니다.
- ✓ 기술은 경험과 지식이 균형을 이룰 때, 빠른 속도로 향상되는 까닭에 새로운 기술에 대한 지식 쌓기는 필수입니다.
- ✓ 이런 환경에서 기술 리드로써 위치를 유지하려면, 아키텍트는 필요한 지식을 스스로 쌓아야 합니다.
- ✓ 정기 교육도 중요하지만 [자기주도 학습(Self-directed learning)] 을 중심으로 지식 목표를 달성해야 합니다.



## 1.4 아키텍트 역량 (3/5) – 창의 (Creativity)

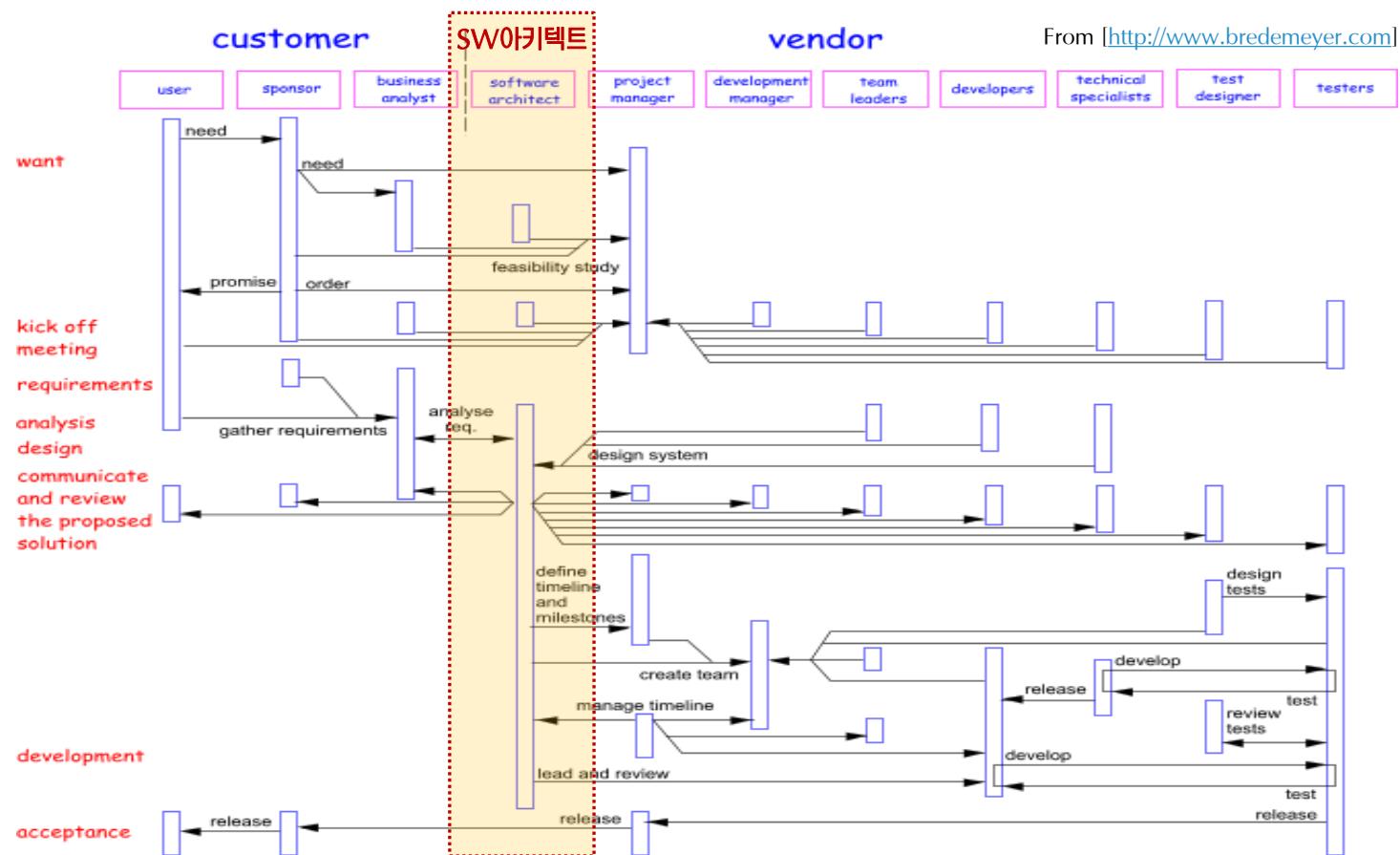
- ✓ 노력은 만족을 가져오지만, 창의적인 아이디어는 감동을 불러일으킵니다.
  - ✓ 기술로 넘지 못하는 벽은 번뜩이는 아이디어로 넘을 수 있습니다. 기술 너머에 아이디어가 있습니다.
  - ✓ 아이디어는 기술로 구현하고, 기술로 넘지 못하는 벽은 아이디어로 넘어갑니다. 그 아이디어를 다시 기술로 구현합니다.
  - ✓ 스마트 시대에는 엔지니어에게 기민함, 탁월함을 요구합니다. 이는 모두 창의성을 바탕으로 합니다.



[출처] <http://www.imindmap.com/articles/files/CreativityTips.jpg>

# 1.4 아키텍트 역량 (4/5) – 소통 (Communication)

- ✓ 설계를 잘 하는 것도 중요하지만, 설계를 둘러싼 소통이 더 중요합니다. 소통이 없으면 설계 구현이 어렵습니다.
- ✓ 인내를 가지고 공유하고 공감하면서 함께 일하여야 합니다. 그러기 위해서 소통(communication)이 중요합니다.
- ✓ 엔지니어링 활동에서는 “대화를 통한 공감대 형성”과 “문서를 통한 명확한 소통”이 균형을 갖추어야 합니다.
- ✓ SW 아키텍트는 프로젝트 의사 소통의 중심 축에 있습니다. 아키텍트의 소통 역량이 프로젝트의 성패를 가늠합니다.



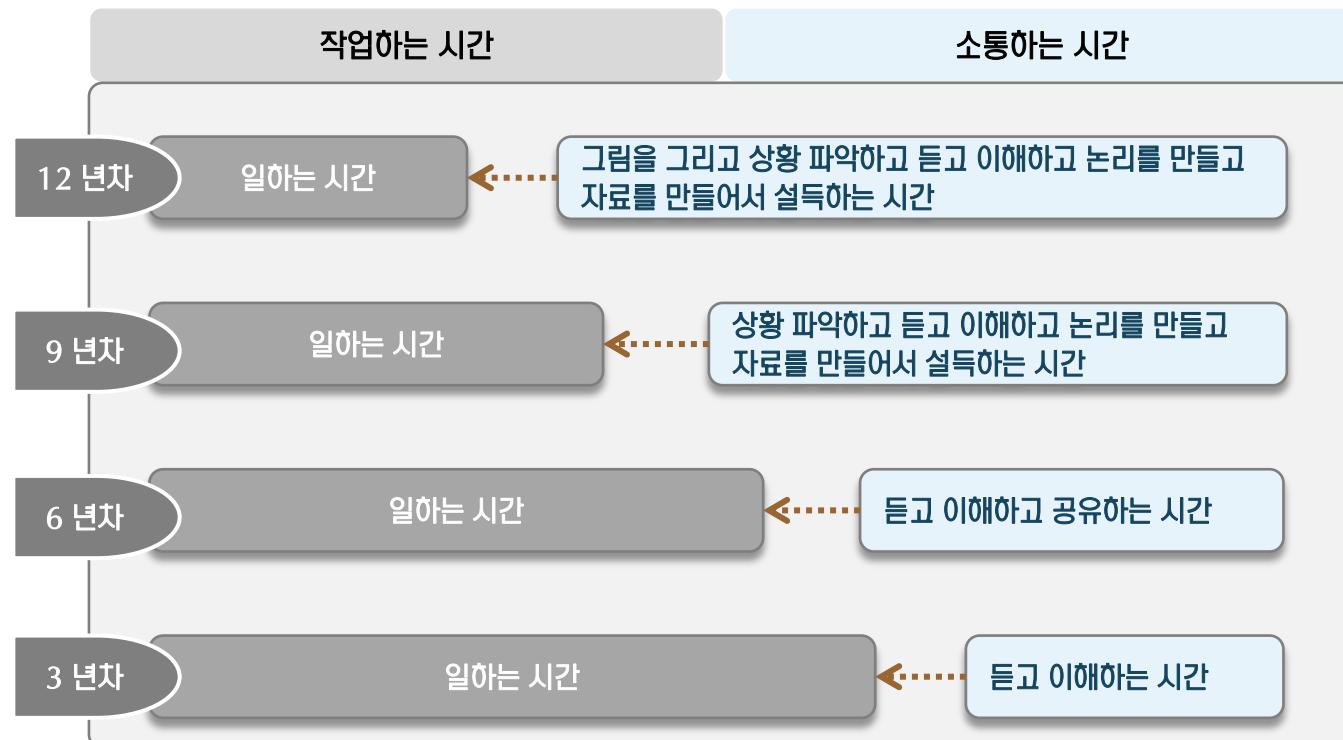
## 1.4 아키텍트 역량 (5/5) – 정신 자세 (Mental attitude)

- ✓ 아키텍트는 개발팀에서 가장 중요한 역할 중에 하나이며, 그에 따른 스트레스 역시 가장 많이 받습니다.
- ✓ 어떤 상황에 처하든지 긍정적인 정신 자세(mental attitude)를 가지고 해쳐나가야 합니다.
- ✓ 프로젝트 상황이 어려울수록, PM의 역할 못지 않게 아키텍트 역할의 중요성은 커 집니다.
- ✓ 이런 자세는 아키텍트가 되었을 때 갑자기 생기는 것이 아니라 엔지니어 경력 경로에서 지속적으로 견고해 져야 합니다.



## 1.5 소통과 작업

- ✓ 주니어(3년 이하) 엔지니어로부터 시니어(12년) 엔지니어에 이르기 까지, SW 개발을 위한 소통을 합니다.
- ✓ 위로 올라 갈수록 실제 자신의 작업을 하는 시간이 줄어들고 소통에 사용하는 시간이 많아집니다.
- ✓ PM, SW 아키텍트, 도메인 모델러(분석가) 등의 역할은 의사 결정을 바탕으로 하는 소통이 주요 활동이 됩니다.
- ✓ 소통을 위한 준비는 주니어 엔지니어로부터 시작되어야 합니다. 글쓰기, 말하기 등을 천천히 준비하여야 합니다.



## 1.6 요약

- ✓ SW 개발은 다양한 역할 담당자들이 서로 다른 기술을 가지고 협업하는 매우 복잡한 활동입니다.
- ✓ 복잡한 세상의 복잡한 비즈니스를 지원하기 위해 애플리케이션은 날이 갈수록 복잡한 모습을 보여 주고 있습니다.
- ✓ 이런 소프트웨어를 개발하는 엔지니어의 기술은 전문화, 세분화되어 갑니다.
- ✓ 따라서, 시간이 흐를수록 SW 엔지니어들에게 중요한 역량은 소통입니다.



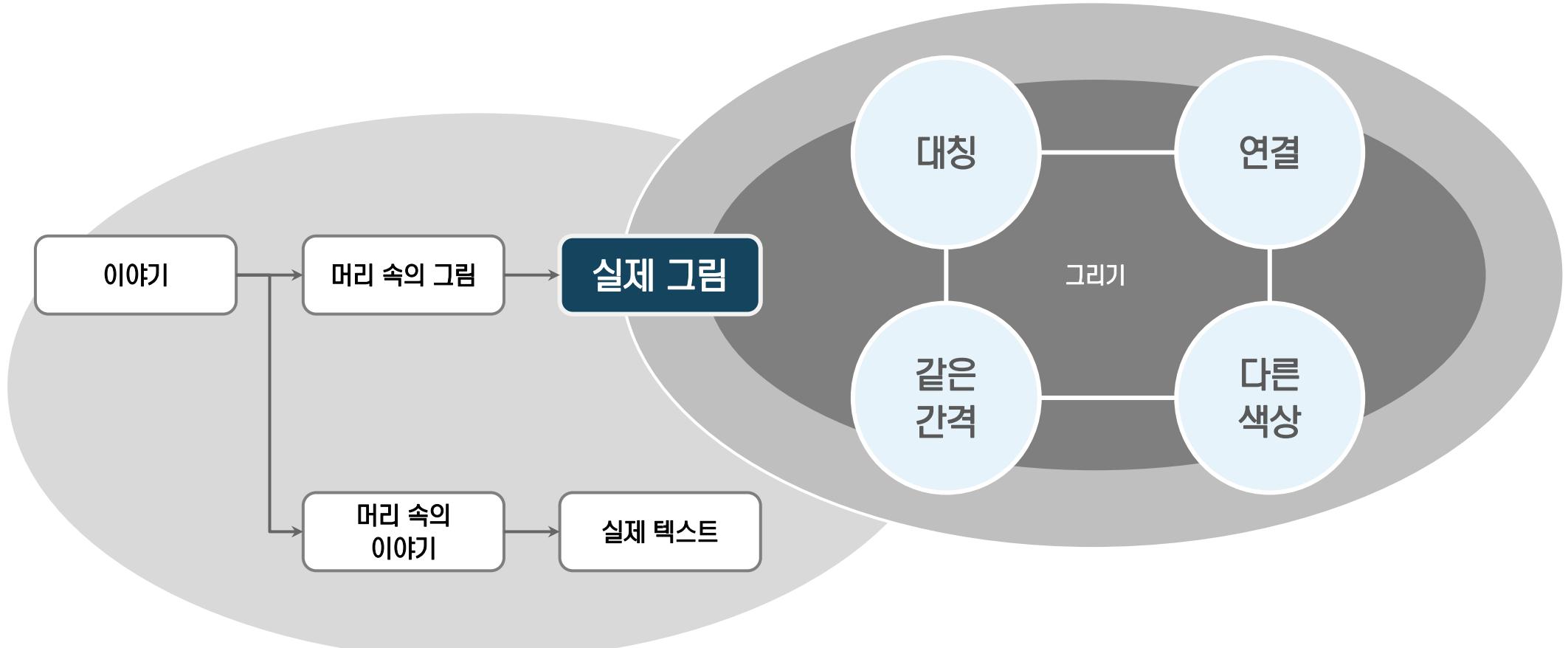


## 2. 표현(representation) 원칙

- 
- 2.1 표현 원칙
  - 2.2 원칙 1 – 정성
  - 2.3 원칙 2 – 시나리오
  - 2.4 원칙 3 – 레이아웃
  - 2.5 원칙 4 – 간격
  - 2.6 요약

## 2.1 표현 원칙 (1/3)

- ✓ 발표 자료를 준비하는 시간은 “생각하는” 시간입니다. ← 절대로 단순 반복 작업이 아닙니다.
- ✓ 글과 그림(도표, 다이어그램 포함)으로 구성되는 발표 자료를 만들 때, 글도 어렵지만, 그림도 어렵습니다.
- ✓ 하지만, 그림은 대칭, 연결, 같은 간격, 다른 색상이라는 네 가지 원칙을 바탕으로 그리면 아주 쉽습니다.
- ✓ 하지만, 이런 원칙에 충실하더라도, 머리 속에서 이야기와 그림이 완성되어야 실제로 그림을 그릴 수 있습니다.



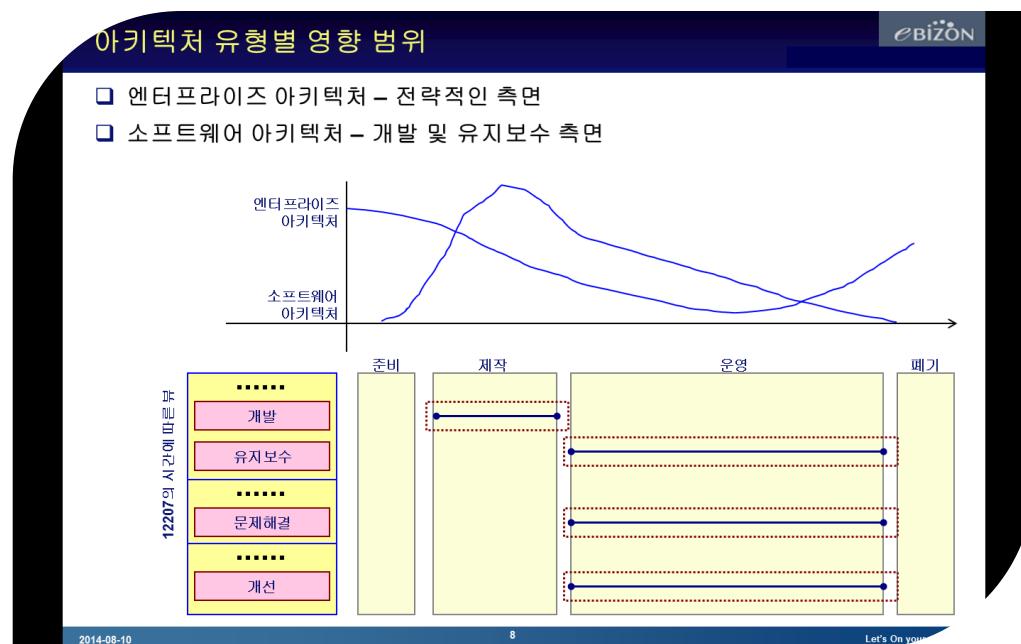
## 2.1 표현 원칙 (2/3)

- ✓ 끝없는 손놀림으로 그려 나가는 세밀화를 보신 적이 있습니까?
- ✓ 손끝에서 탄생하는 놀라운 그림들이 단순 반복 작업의 결과로 보이시나요?
- ✓ 붓과 펜이 도화지를 스쳐 지나갈 때마다, 화가의 마음 속에 있는 느낌들이 화폭으로 쏟아져 들어옵니다.
- ✓ 발표 자료에서도 마찬가지입니다. 도형을 선택하고 옵션을 설정하는 과정은 “경험과 지식을 담아 내는 과정”입니다.



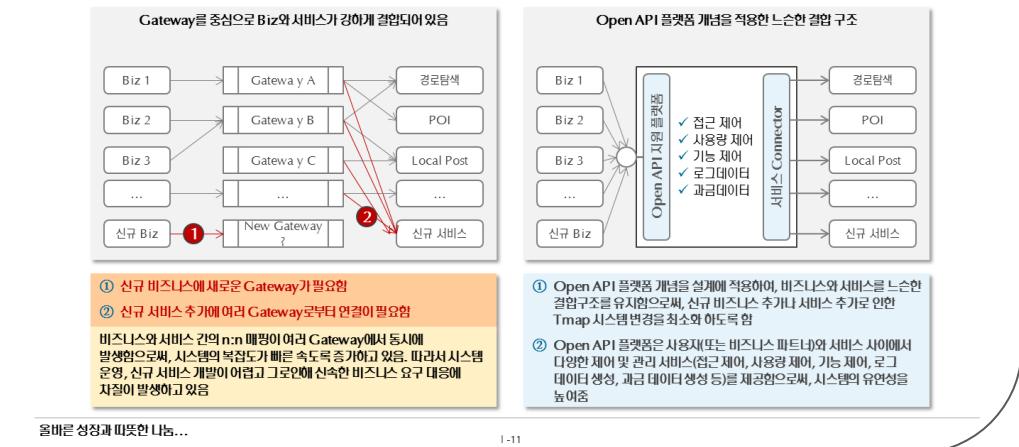
## 2.1 표현 원칙(3/3)

- ✓ 많은 시간 정성을 들여 발표 자료를 만들면, 들인 시간 만큼의 충분한 가치가 있습니다.
- ✓ 잘 만든 자료는 내용과 마음이 함께 전달됩니다. 그래서 설득력이 있습니다. 자료의 목표를 바로 달성할 수 있습니다.
- ✓ 잘 만든 자료는 오래오래 사용됩니다. 다른 사람들이 참조를 통해 재사용할 수 있습니다.
- ✓ 자료는 표현에 들이는 시간 만큼의 가치를 발휘합니다.



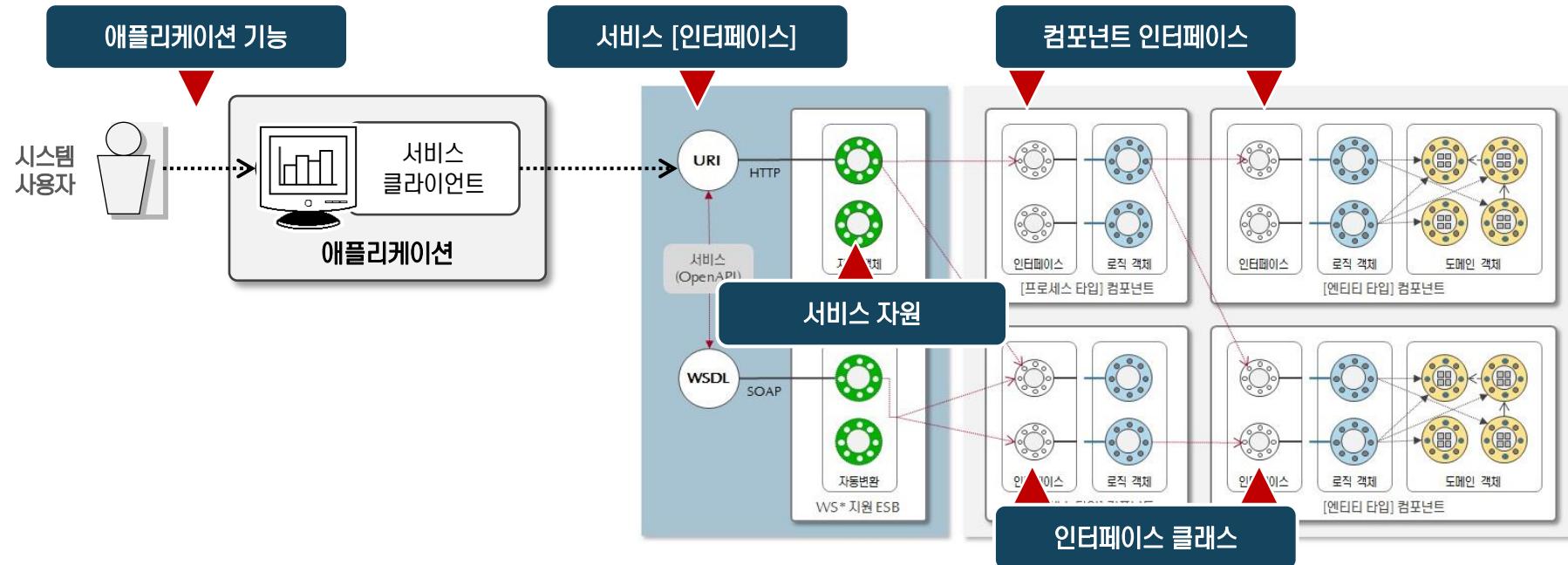
### 2. Demo 시스템 개선방향

- ✓ 차세대 Demo 재개발 워크그룹이 식별한 가장 심각한 문제는 Gateway 주변의 복잡도 증가임
- ✓ 여러 Gateway에 형성되는 N:N 관계를 OpenAPI 플랫폼 중심의 N:1:N의 느슨한 관계로 풀어나감
- ✓ 향후 Biz도 등록하고 서비스도 등록한 후, 둘 간의 관계를 구독(subscription)으로 풀어 가야함



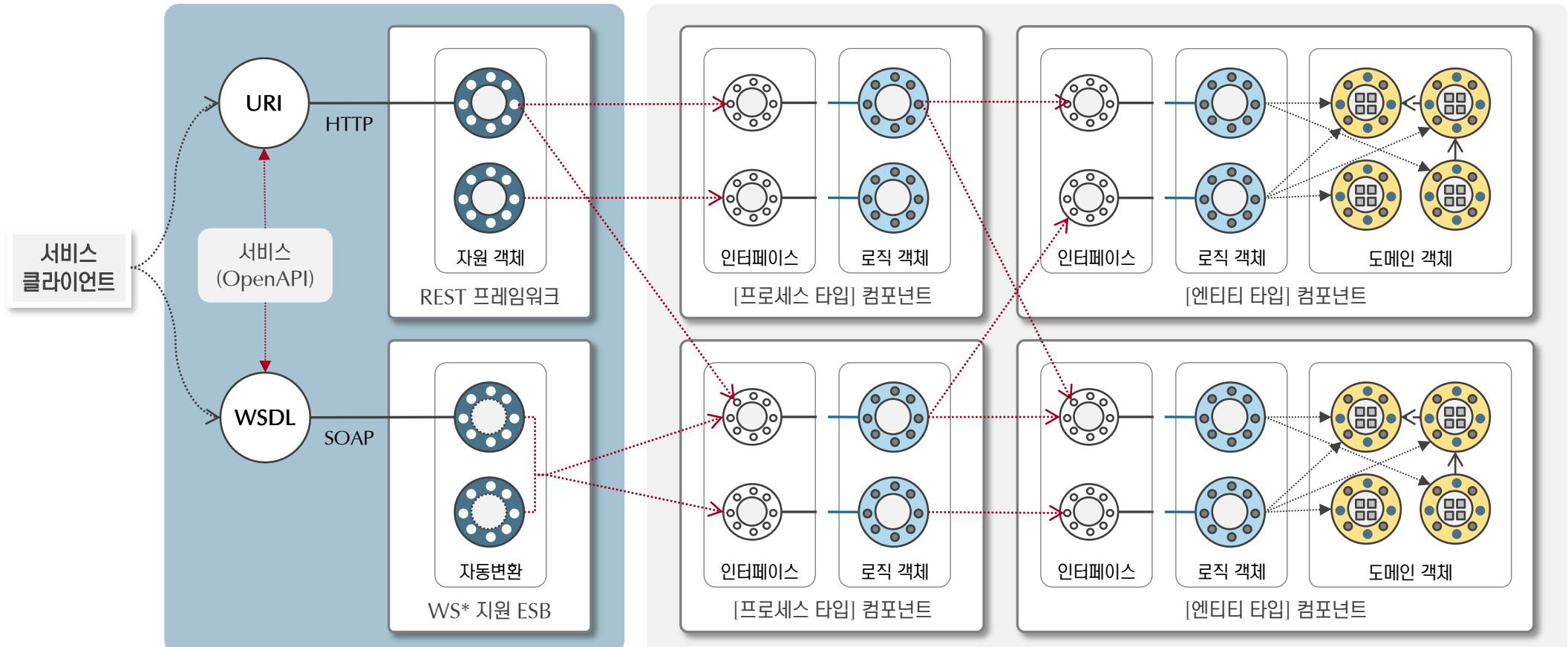
## 2.2 원칙 1 – 정성 [1/2]

- ✓ 상황이 복잡할수록, 의견의 차이가 클수록, 보다 정확한 표현이 필요합니다.
- ✓ SOA(Service Oriented Architecture)의 개념으로 인해 혼란스러운 프로젝트에 참여한 적이 있습니다.
- ✓ 많은 사람들이 “도대체 서비스가 뭐냐...”라며 논쟁에 논쟁을 거듭하고 있었습니다.
- ✓ 논쟁은 6개월 간 계속되었고, 결국은 프로젝트를 벼랑 끝으로 몰고 갔습니다.



## 2.2 원칙 1 – 정성 [2/2]

- ✓ 객체, 컴포넌트, 서비스 이 세 가지 설계 요소가 서로 어떤 관계가 있는지 설명할 필요가 있었습니다.
- ✓ 적당한 표현은 서로 다른 생각을 가진 사람들 간에 혼란을 부추길 수 있는 상황입니다.
- ✓ 기술적인 내용이지만 기술을 모르는 이해관계자(stakeholder)도 설득할 수 있도록 자료를 준비했습니다.
- ✓ 다음 그림은 논쟁의 핵심인 “SOA에서 서비스의 정체”를 밝히는데 충분한 자료가 되었습니다.



## 2.3 원칙 2 – 시나리오 (1/2)

- ✓ 시나리오는 이야기입니다. 이해관계자들이 듣고 싶어하는 내용을 어떤 순서로 이야기해 줄까를 정합니다.
- ✓ 초기에 전체 시나리오를 완성하고 자료를 만드는 경우는 별로 없습니다. 중요한 자료 순으로, 또는 만들기 편한 자료 순으로 자료를 만들어 가다 보면, 부분적으로 완성된 자료를 어떻게 연결해야 할지 아이디어가 떠오릅니다.
- ✓ 발표자가 하고 싶은 이야기와 청중이 듣고 싶은 이야기가 잘 조화를 이루어야 합니다.

시나리오 예제 #1

1. 컨설팅 개요
2. 인터뷰
3. 역할(Role)
4. 기술(Skill)
5. 로드맵(Roadmap)
6. 역량 분석(Assessment)
7. 교육훈련(Training)
8. 로드맵 실행

시나리오 예제 #2

1. T map 시스템 아키텍처 목표
2. T map 시스템 개선방향
3. 아키텍처 요구사항
4. 아키텍처 설계전략
5. 시스템 식별
6. 언어 및 플랫폼
7. 시스템 구조 설계
8. Open API 설계
9. End-to-End 흐름구조

시나리오 예제 #3

1. 기업의 IT 영역
2. 비즈니스 분석 컨텍스트
3. 비즈니스 모델링 리뷰(BPR, ARIS, PWC)
4. 비즈니스 모델링의 목표
5. 비즈니스 모델링 트랜드
6. IT 기술 트랜드
7. IT 기술 트랜드와 개발방법론
8. [비즈니스 분석영역에서] 서비스 개념
9. 요약

## 2.3 원칙 2 – 시나리오 (2/2)

- ✓ 좋은 시나리오를 얻기 위해서는 많은 생각이 필요합니다. ← 시나리오는 지식보다는 논리를 바탕으로 하는 생각입니다.
- ✓ 초기에 스케치 북을 이용하면 생각의 흐름을 정리하는데 많은 도움이 됩니다.
- ✓ 초기부터 애플리케이션 도구를 사용하면, 생각의 흐름을 정리하는 것이 아니라 도구와 싸움으로 흐르는 경우가 많습니다.
- ✓ 스케치 북에서 이야기의 흐름이 완성되거나 아이디어가 구체화 되면, 도구로 옮겨서 작업을 합니다.

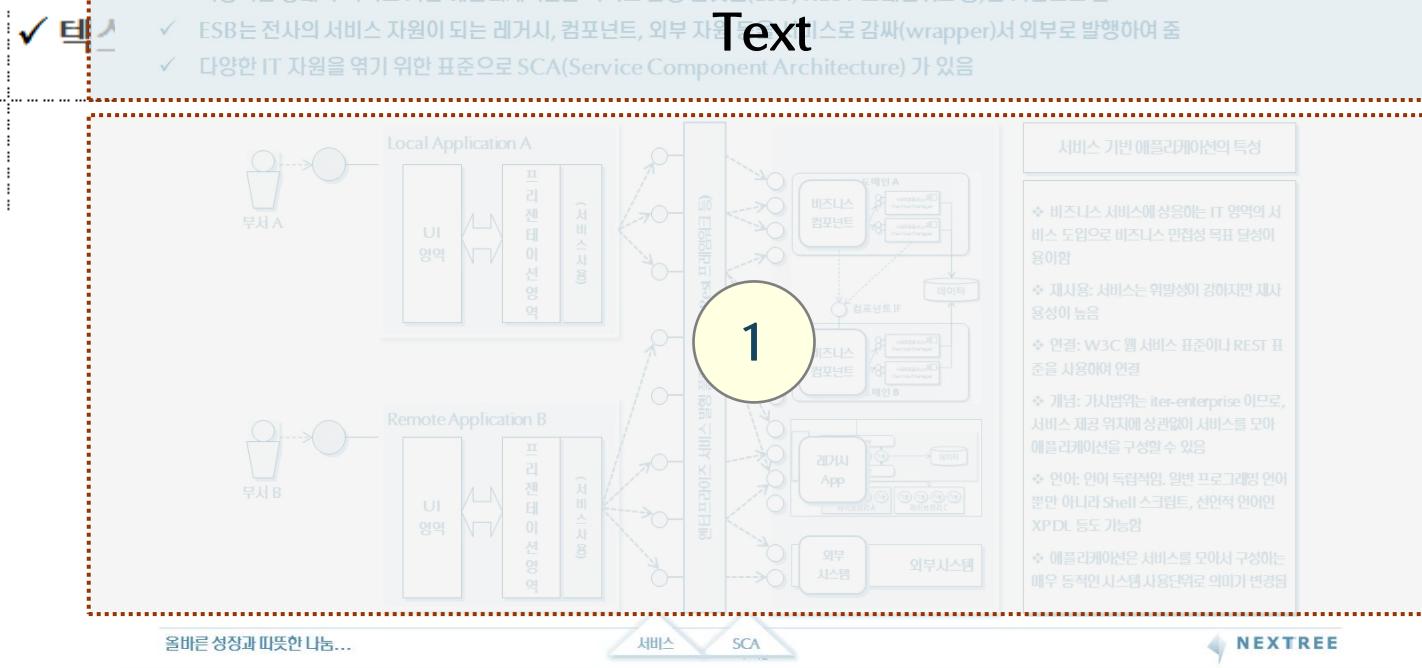
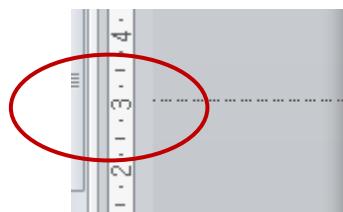


## 2.4 원칙 3 – 레이아웃 (1/2)

- ✓ 레이아웃은 페이지의 아키텍처입니다. 전체 페이지에 걸쳐서 가급적 통일된 구조를 가져가야 합니다.
- ✓ 레이아웃이 없을 경우, 컨텐츠의 위치는 불규칙하여, 페이지를 전개할 때 혼란스러운 느낌을 줍니다.
- ✓ 전체 페이지의 70% 이상을 팀의 기본 페이지 레이아웃을 사용하면 표현의 안정감을 높일 수 있습니다.
- ✓ 텍스트와 다이어그램/그림 모두 각자의 규칙을 가져야 합니다. 텍스트는 가급적 라인 수를 채우고, 그림은 균형을 갖춥니다.

### 1. SOA 이미지 (2/2)

posco



## 2.4 원칙 3 – 레이아웃 (2/2)

- ✓ 비교하거나 대안을 제시할 경우, 좌측 하단의 레이아웃을 사용합니다.
  - ✓ 다이어그램이나 그림이 클 경우, 우측 하단의 레이아웃을 사용합니다. 어떤 경우든, 텍스트가 포함되는 것이 좋습니다.
  - ✓ 다이어그램이나 그림만으로 페이지를 채우면 안됩니다. 텍스트를 포함하여 내용을 이해하기 쉽도록 해야 합니다.
  - ✓ 미리 레이아웃을 그려 놓고 도형이나 다이어그램을 채우는 것 보다는, 다 그려 놓고 조정하는 것이 효율적입니다.

## 1. SOA 이미지 (2/2)

posco

- 이상적인 상태의 서비스 기반 애플리케이션은 서비스 활용형(ESB, REST 프레임워크 등)을 기반으로 함
- ESB는 전사의 서비스 자원이 되는 레거시, 컴퓨터트, 외부 자원 등을 서비스로 감싸(wrapper)서 외부로 발행하여 줌
- 다양한 IT 자원을 역할 위핚 표준으로 SCA(Service Component Architecture)가 있음

### Text

The diagram illustrates the evolution of SOA through three stages:

- 1. Local Application A:** Shows a monolithic application structure with a UI layer and business logic layers. It is labeled "부서 A".
- 2. Service-oriented architecture (SOA):** Shows the application split into multiple services (e.g., Business Logic 1, Business Logic 2) connected via a Bus (e.g., ESB, REST API). It is labeled "부서 B".
- 3. Service Component Architecture (SCA):** Shows the services integrated into a single composite application (e.g., App) using SCA. It is labeled "부서 C".

Annotations on the right side of the diagram provide additional context:

- 수 베이스 서비스에 상관없이 IT 영역의 서비스 도입으로 베이스는 면밀성 확보 달성이 가능함
- 수 대시보드 서비스는 위임성이 강하자면 제작 품질이 높음
- 수 면밀-W3C 웹서비스 표준과 REST 표준을 사용하여 연계
- 수 개별 서비스별로 서비스 관리가 가능하여 관리가 간편화된다.
- 수 면밀 면밀 서비스는 면밀 서비스로 표준화된 면밀이나 Shell 스크립트로 표준화된 XML로 등록 가능함
- 수 웹플랫폼은 서비스를 모아서 구성하는 데 매우 유동적인 시스템 사용자에게 어려움이 없도록

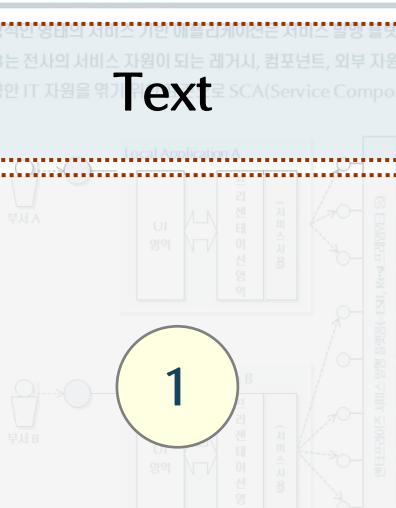
출처: 출처: 출처: 출처: 출처:

NEXTREE

1. SOA 이미지 [2/2]

- 이상적인 형태의 서비스 기반 애플리케이션은 서비스 일정을 갖음
- ESB는 전사의 서비스 지원이 되는 레거시, 컴포넌트, 외부 지원 등을 서비스로 감싸(wrapper)서 외부로 발행하여 줌
- 다양한 IT 지원을 위한 규칙으로 SCA(Service Component Architecture) 가 있음

**Text**



2. SOA 구조 [2/2]

2b. REST 프레임워크 등)을 기반으로 함

서비스 기반 애플리케이션의 특성

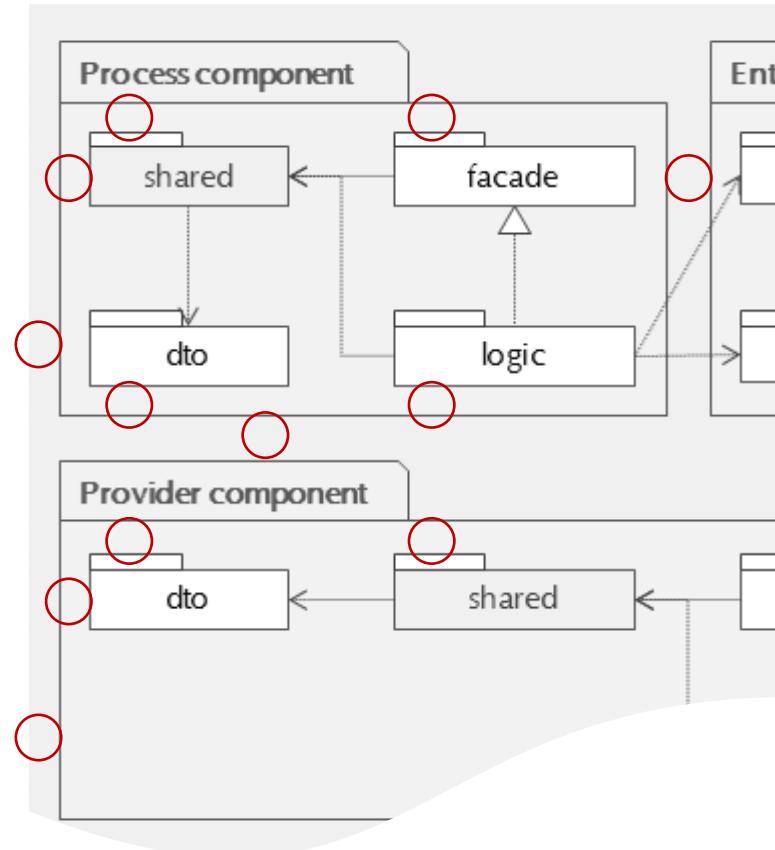
- 서비스는 서비스 제공자는 IT 영역에서 서비스 도입으로 서비스 연동을 목표 달성이 목표이다.
- 서비스는 기존에 강화되었지만 재사용성이 확장되었고, 재사용성이 확장되었다.
- for-enterprise 이므로, 서비스 제공 기관에 상관없이 서비스로 모아 애플리케이션을 구현할 수 있다.
- 언어: 언어 독립적인 맵핑 프로토콜(언어 분리)이나 Shell 스크립트, 산업적 언어인 XML 등도 가능함
- 애플리케이션은 서비스를 모아서 구성하는 매우 등식적 시스템 사용으로 의미가 한정됨

**2**



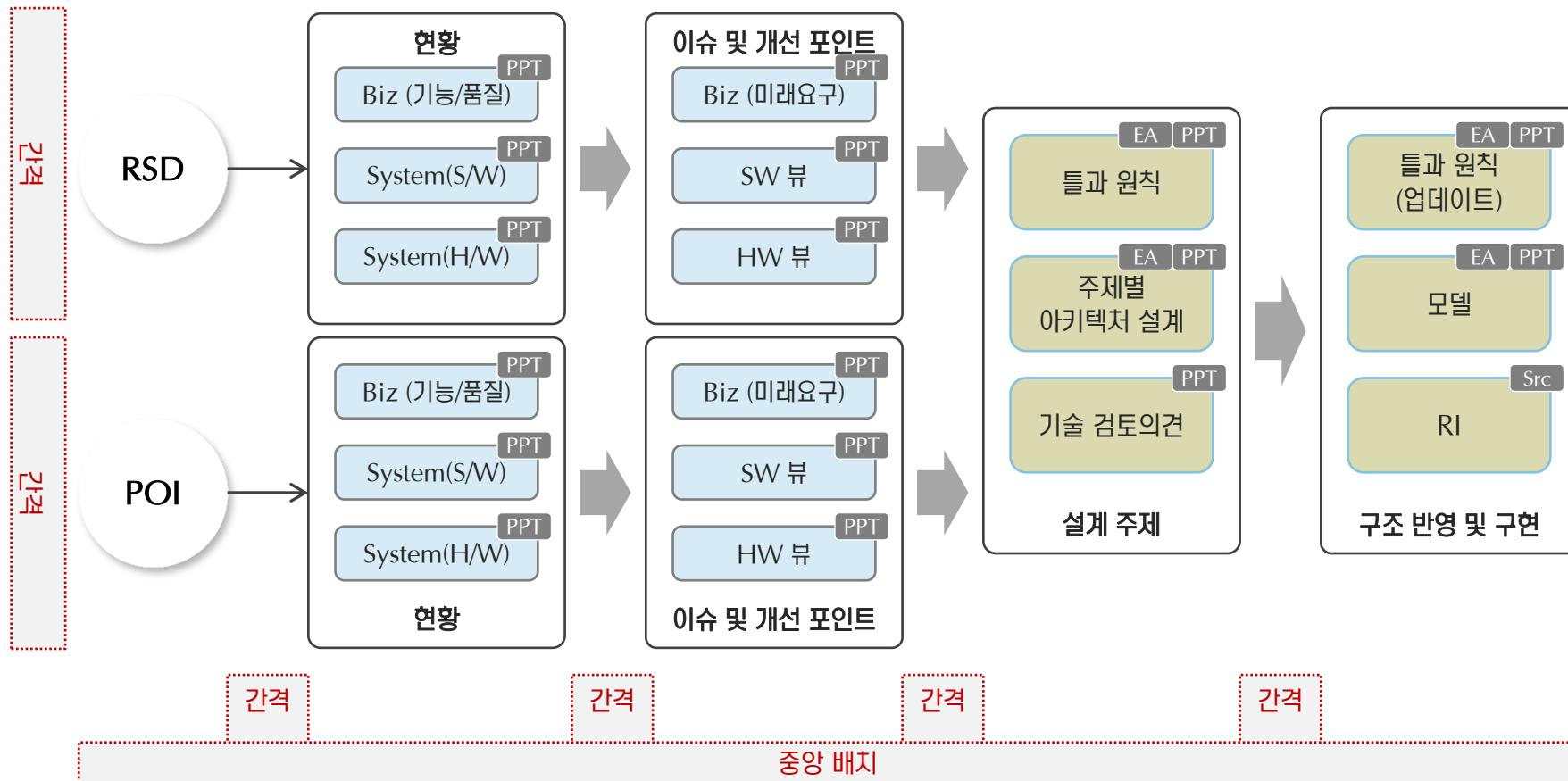
## 2.5 원칙 4 – 간격 (1/2)

- ✓ 도형과 다이어그램의 간격은 전체적인 균형을 제공함으로써, 안정적일 뿐만 아니라 아름답게 보이게 합니다.
- ✓ “동일 간격의 원칙”을 준수하면 단정한 모양을 유지하여, 눈에 잘 들어올 뿐만 아니라 정보를 더 잘 전달할 수 있습니다.
- ✓ 어떤 곳이든 간격이 있는 곳은 대등한 요소를 찾아서 “동일 간격”을 유지하도록 노력합니다.
- ✓ 눈금 및 안내선 화면에서 눈금 설정 값을 팀에서 공유하지 않으면, 팀 작업에서 간격을 맞추기 어려우므로 주의합니다.



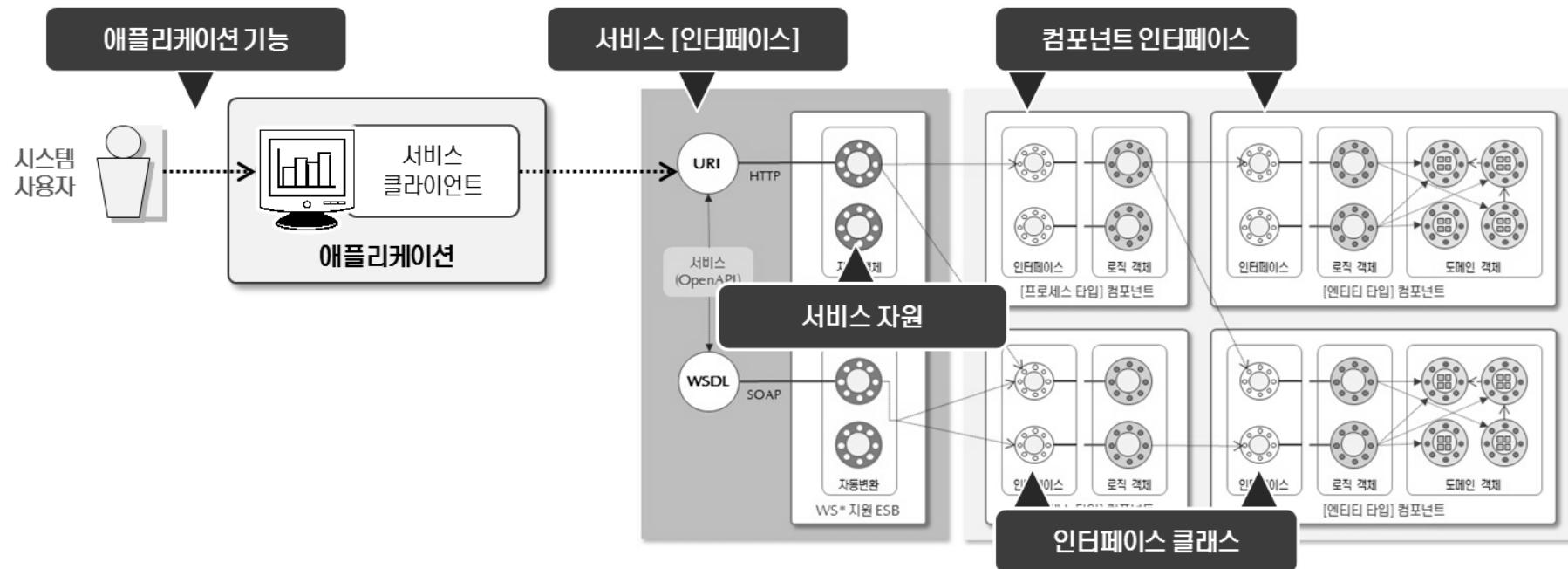
## 2.5 원칙 4 – 간격 [2/2]

- ✓ “동일 간격의 원칙”은 도형, 다이어그램, 테이블 등 어떤 요소든 가리지 않고 적용할 수 있습니다.
- ✓ 좌우 여백도 “동일 간격의 원칙”을 적용해야 합니다. 따라서 그림의 테두리가 없더라도 가상의 테두리를 가정해야 합니다.
- ✓ 아래 화살표는 도형의 한 가운데 위치함으로써 상하로 동일 간격을 유지할 수 있습니다.
- ✓ 심지어는 아래 그림의 RSD 원과 도형 간의 간격도 유사한 도형 간의 간격과 동일하게 유지합니다.



## 2.6 요약

- ✓ 표현 원칙은 대단한 것이 아니라, 그저 표현하는 스타일을 유지하기 위한 규칙의 집합입니다.
- ✓ 누구나 지속적으로 표현 훈련을 하면서 나름의 원칙을 만들어 갈 수 있습니다.
- ✓ 그리고 조직에서 표현 원칙을 정의하고 공유한다면 많은 효과가 있을 수 있습니다.
- ✓ 조직의 표준 템플릿을 가지고 있다면 더 할 나위 없이 좋겠습니다. 대신 템플릿도 필요한 최소한의 정의가 미덕입니다.





### 3. 실습 #1 - 다른과 같음

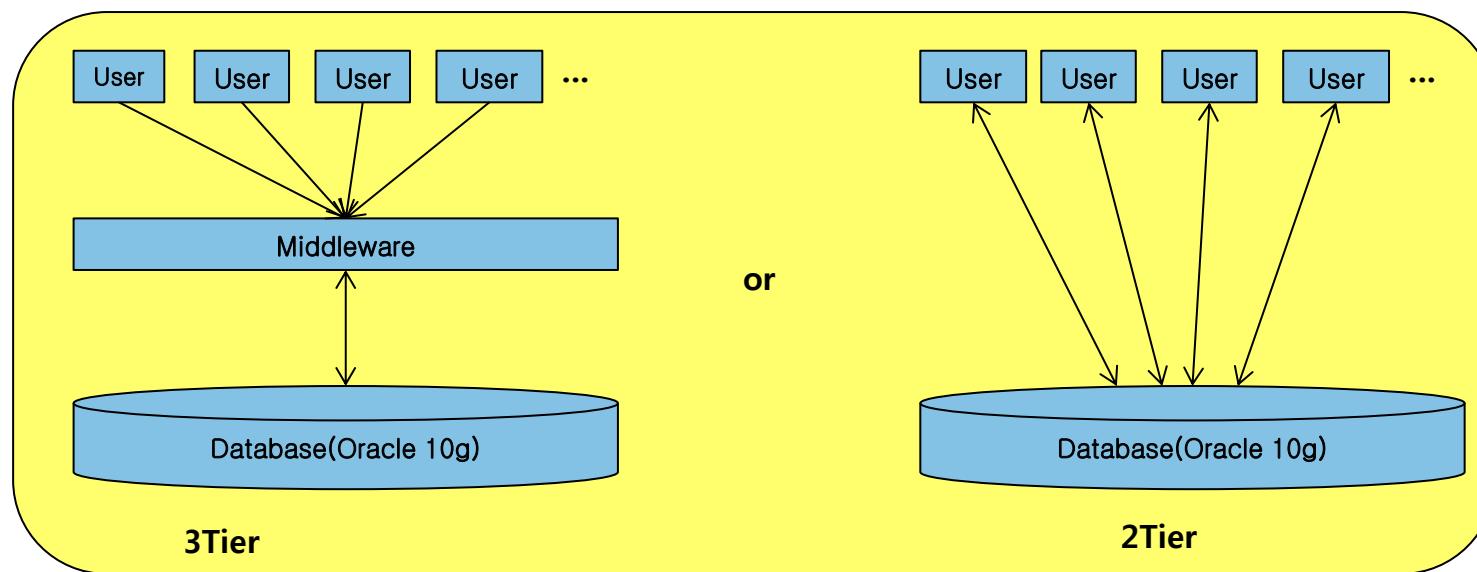
- 
- 3.1 개발자 다이어그램
  - 3.2 개선 유형 1 – 간결함
  - 3.3 개선 유형 2 – 색상, 입체감
  - 3.4 그리기 안내
  - 3.5 그리기
  - 3.6 요약

## 3.1 개발자 다이어그램

- ✓ 어느 개발자 분이 작성한 문서의 한 부분입니다. 시스템 아키텍처 중에 티어 구조를 설명하고 있습니다.
- ✓ 기본 도형, 기본 선, 원색 등으로 구성한 시스템 구성도입니다.
- ✓ 무슨 내용인지 알겠으나, 이런 식으로 설계를 한 설계자에 대한 믿음이 가지 않습니다.
- ✓ 마찬가지로, 이런 자료를 관리자 분들이나 고객 분들께 제시했다가는 무슨 일을 당할지 모릅니다.

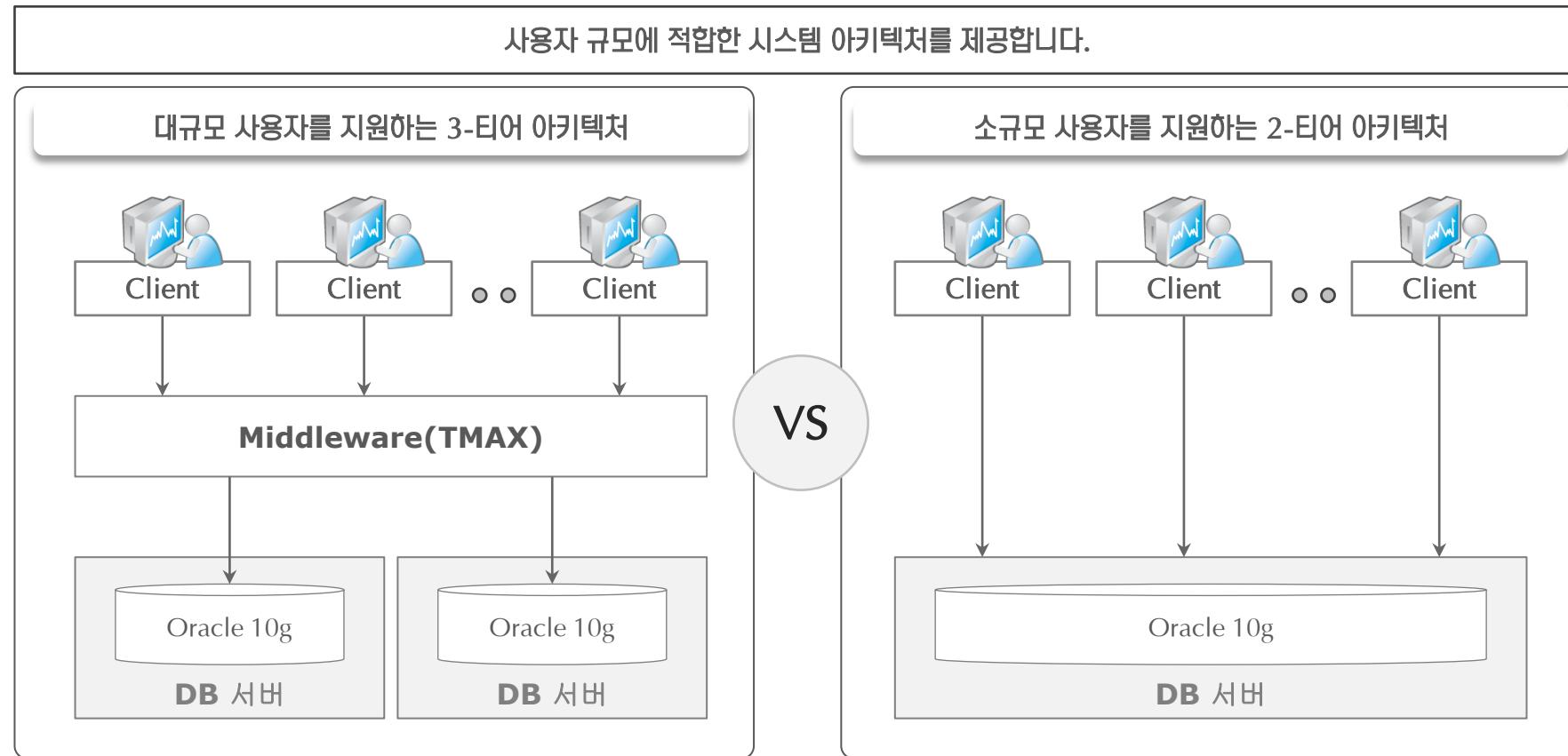
함께 그려 보실까요?

사용자 규모에 적합한 연결방식 사용



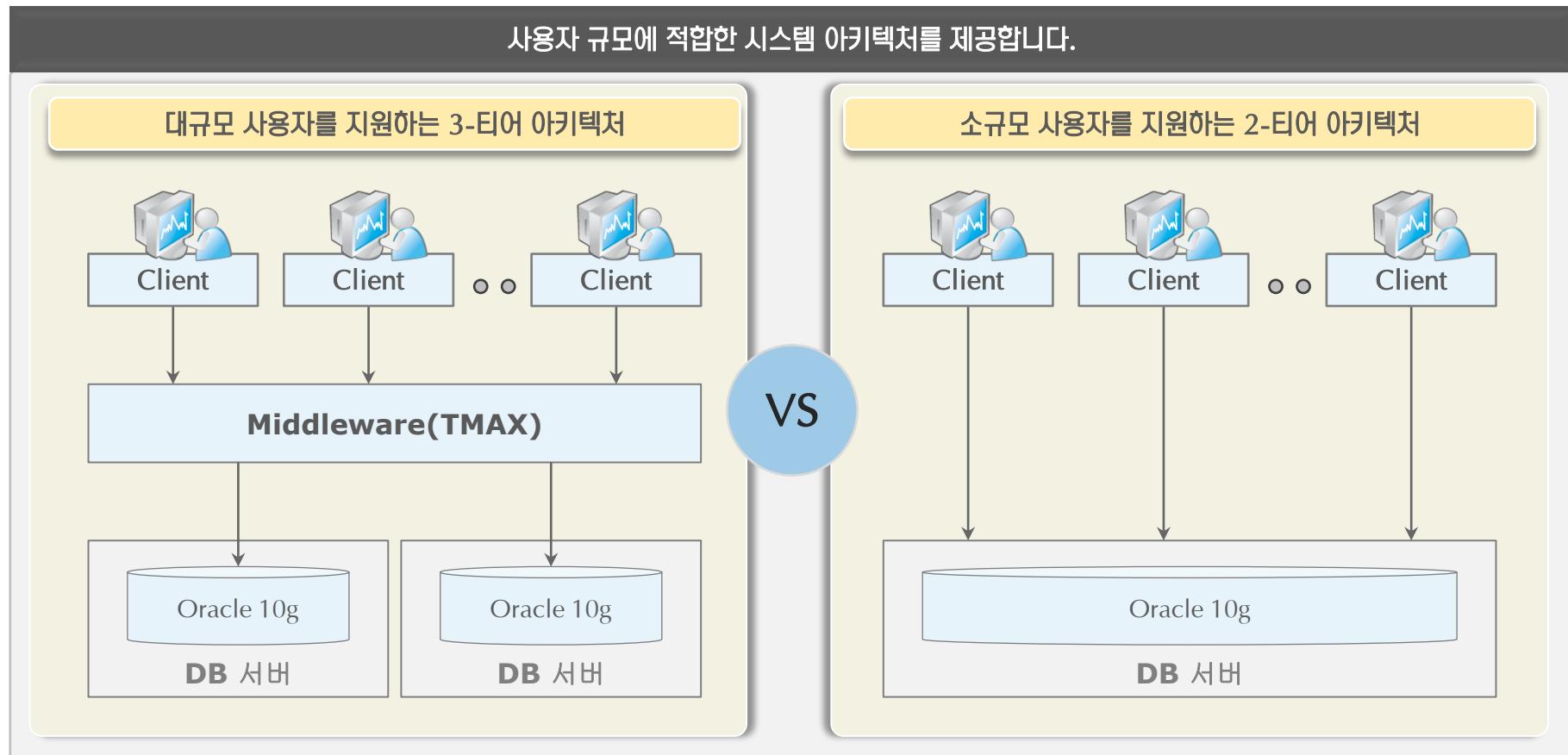
## 3.2 개선 유형 1 – 간결함

- ✓ 선, 도형, 색상, 폰트 등을 조정함으로써 의미 전달이 잘 되도록 하였습니다.
- ✓ 물론 미학 관점에서 이런 방식의 도형이 정말 전달력이 있는가에 대한 검증은 별개의 문제로 봅니다.
- ✓ 시스템의 이해관계자(stakeholder) 분들은 이러한 설계도를 빨리 그리고 정확하게 이해를 할 수 있습니다.
- ✓ 무엇보다고 설계 엔지니어의 정성과 진심을 느낄 수 있습니다. ← 그리는 시간은 생각하는 시간입니다.



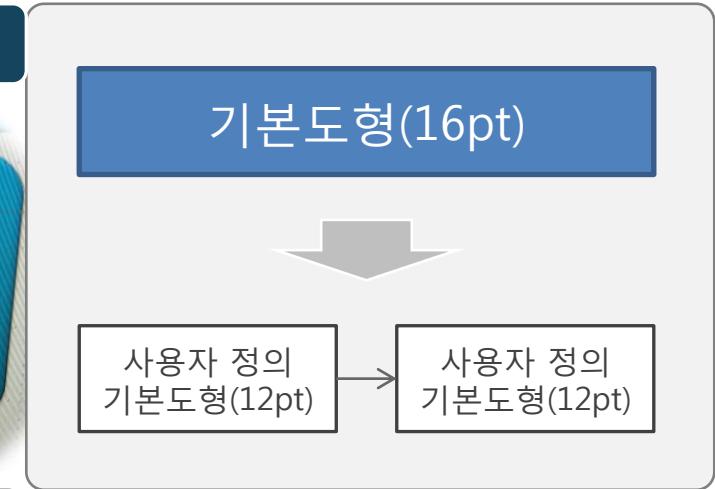
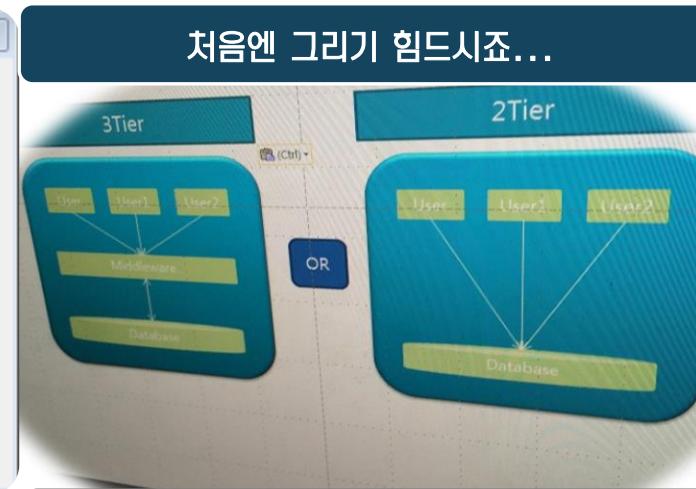
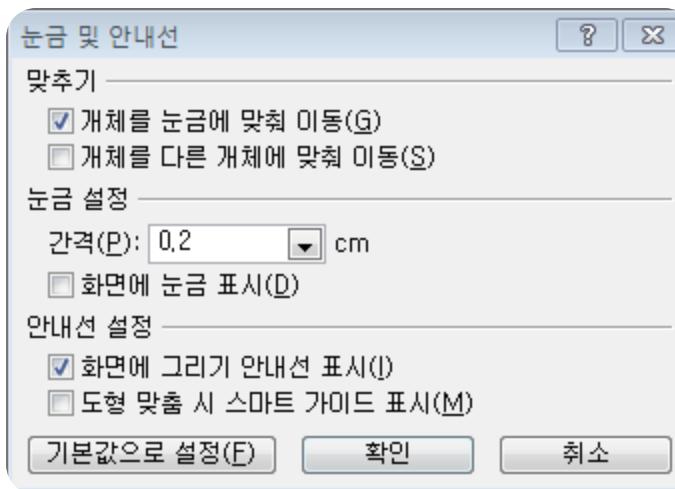
### 3.3 개선 유형 2 – 색상, 입체감

- ✓ 제안서 등과 같이 외부에서 프리젠테이션을 할 때 사용하는 경우에는 색상이나 입체감을 부여합니다.
- ✓ 색상은 전체 자료에서 통일감을 부여하여야 하므로 신중하게 색상을 선택하여야 합니다.
- ✓ 입체감은 필요한 곳에 최소한으로 사용하는 것이 좋습니다. 너무 입체, 음영 등을 사용하면 혼란스러운 그림이 됩니다.
- ✓ 도형을 사용하는 규칙을 전체 자료에 일관되게 적용하여야 합니다.



## 3.4 그리기 안내(1/2)

- ✓ 기본 설정 상태에서 자료를 만들면, 사용하기에 불편한 설정입니다.
- ✓ 눈금 및 안내선을 아래와 같이 조정합니다. 특히, 눈금 간격은 팀에서 동일한 크기로 설정합니다.
- ✓ 기본 도형을 현실적인 크기(12pt 내외)로 변경하고, 폰트도 자주 사용하는 것을 변경한 후, “기본 도형으로 설정”합니다.
- ✓ 기본선도 현실적인 색상, 두께로 변경하고 “기본 선으로 설정”합니다.

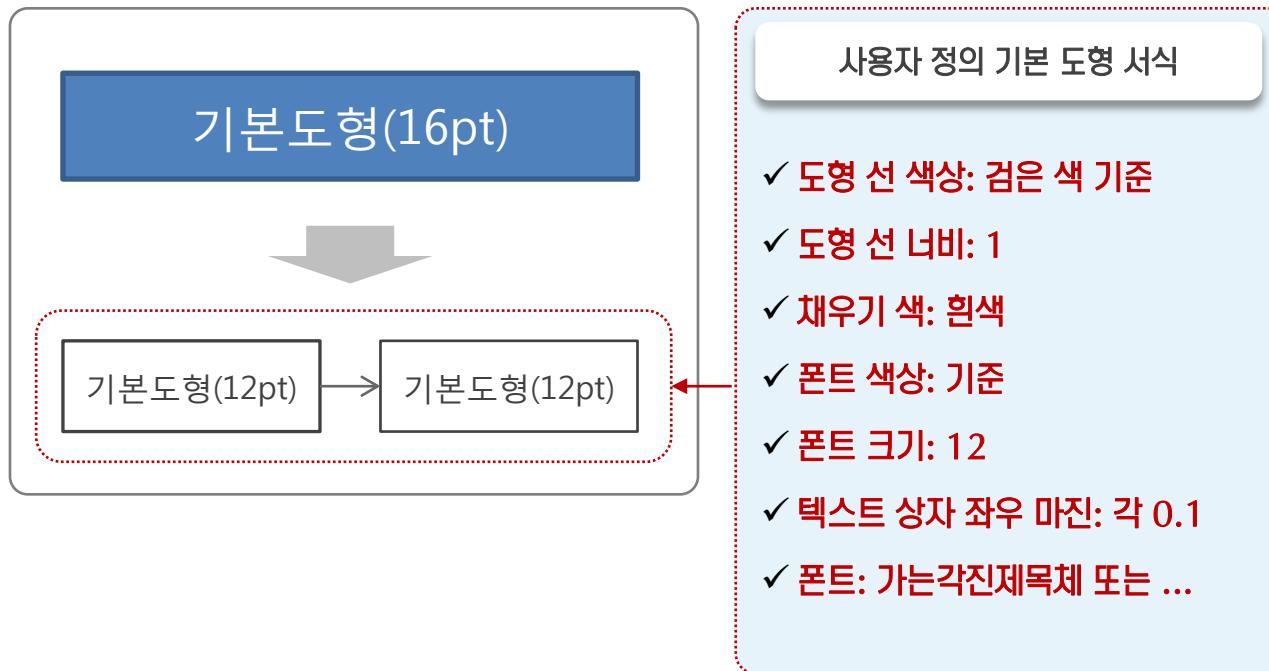


색상: 흰색 바탕에 검은색 선  
두 개의 사각형과 선에 집중  
이후 복사를 통해서 완성

색상과 싸움하지 마라...

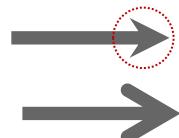
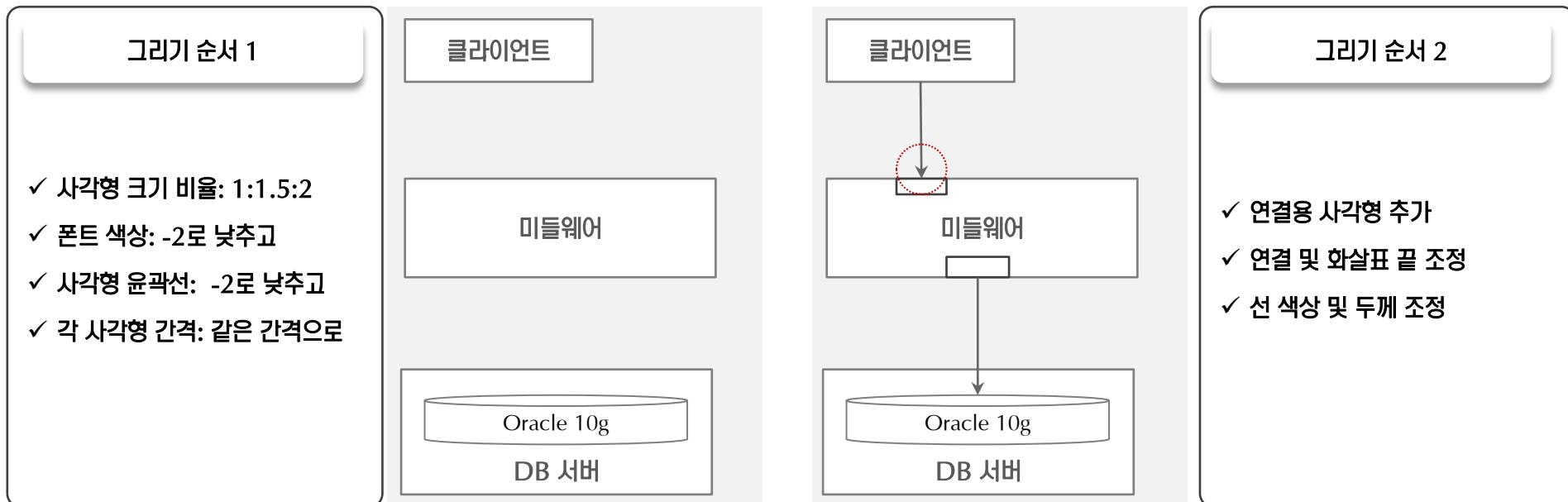
## 3.4 그리기 안내(2/2)

- ✓ 프리젠테이션 도구에서 설정한 도형의 기본 서식을 사용자가 서식의 기본 값을 정의하여 사용합니다.
- ✓ 선의 너비(두께)와 색상, 폰트의 크기와 색상, 채우기 색상, 폰트 종류 등을 선택합니다.
- ✓ 모든 색의 기준은 “검은 색, 25% 더 밝게”로 설정합니다. 앞으로 이 색상을 “기준색”이라 부릅니다.
- ✓ 검은 색을 이야기 할 때, 이 기준 색으로 부터 한 칸 더 밝게는 -1, 더 어둡게는 +1과 같은 방식으로 부릅니다.



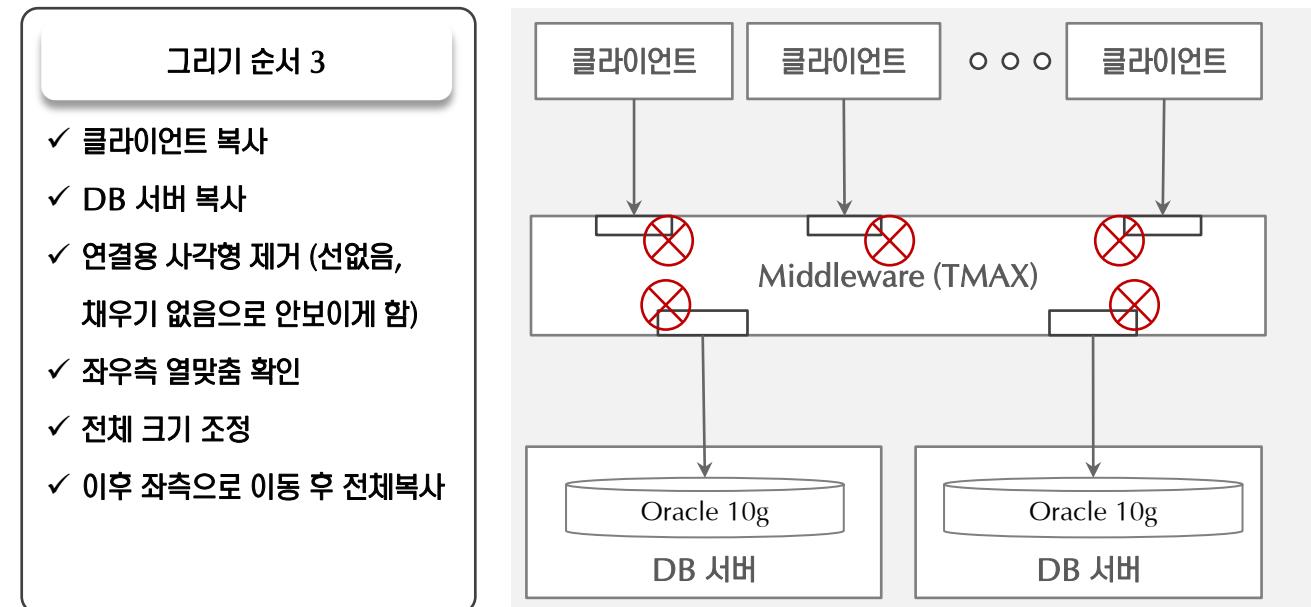
## 3.5 그리기(1/2)

- ✓ 기본 설정이 끝나면 사각형 하나를 그리는 것으로 그리기를 시작합니다. 처음 “기본 세트” 그리기를 합니다.
- ✓ 사각형 하나를 그리고, 폰트, 선 두께, 선 색상을 조정한 후에, 그 사각형을 기준으로 같거나 몇 배 큰 사각형을 그립니다.
- ✓ 사각형 간의 간격을 맞추고, 사각형을 연결선을 이용하여 연결합니다.
- ✓ 연결을 위한 사각형 또는 원형을 사용합니다. 후에 “선 없음, 채우기 없음”으로 보이지 않게 조정합니다.



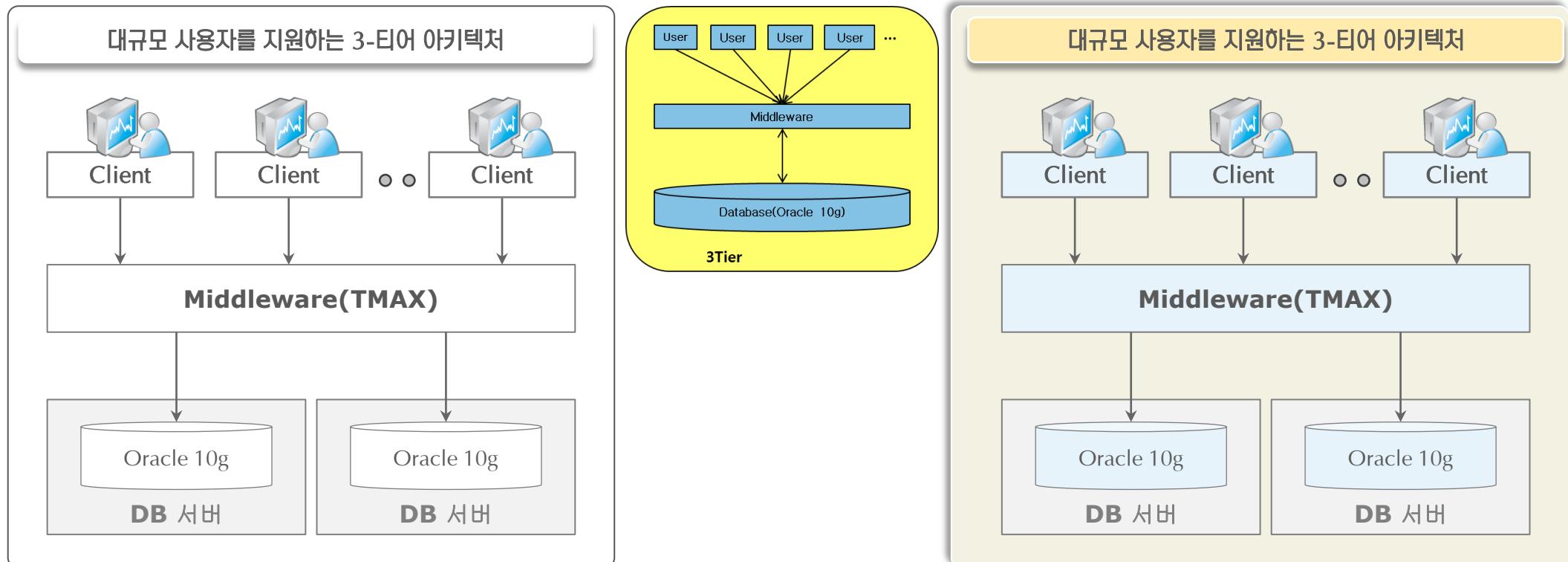
## 3.5 그리기(2/2)

- ✓ “기본 세트” 그리기가 끝나면 다음은 복사하여 붙이기를 통해 필요한 개수 만큼 도형 그룹을 만들어 냅니다.
- ✓ 좌우측 맞춤을 확인하고, 세밀한 간격 조정을 합니다. → 간격 조정할 때 전체적으로 간격을 조정할 수도 있습니다.
- ✓ 모든 조정이 끝나면, 연결용 사각형을 “선없음, 채우기 없음”으로 보이지 않게 조정합니다.
- ✓ 이제 좌측 세트가 끝났으므로, 위치를 왼쪽으로 이동한 후, 이 세트를 복사하여 우측 세트를 만들어 냅니다.



## 3.6 요약

- ✓ 같은 내용의 그림이지만, 어떻게 표현하는가에 따라 정보 전달이나 설득에는 큰 차이가 있습니다.
- ✓ 별 생각 없이 그린 다이어그램이 작성자가 의도하는 목표를 달성하기는 어렵습니다.
- ✓ 그리는 시간은 표현하는 시간으로, 생각하는 시간입니다. 그리는 시간이 길어진다고 불평할 이유가 없습니다.
- ✓ 들인 시간에 대한 효과는 **설득력, 재사용 및 참조** 등 뿐만 아니라, 고민하는 “**시간의 길이만큼 충실한 결과**”입니다.





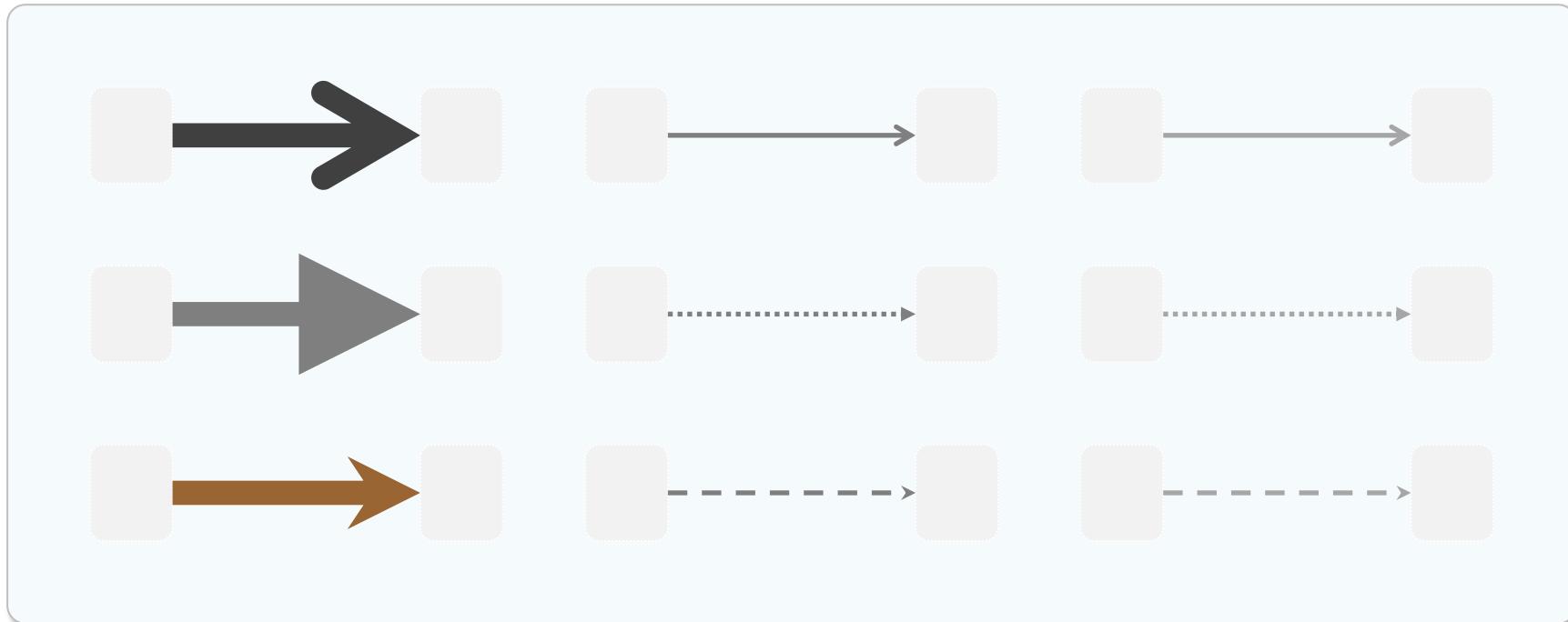
## 4. 기본 그리기

---

- 4.1 선
- 4.2 다각형
- 4.3 선 연결
- 4.4 선 색
- 4.5 다각형 모양
- 4.6 도형 변경
- 4.7 도표
- 4.8 텍스트 상자
- 4.9 자유 도형
- 4.10 요약

## 4.1 선

- ✓ 선은 프리젠테이션 그림의 기본 그림 요소로 선과 도형 모두에 적용됩니다.
- ✓ 기본 선을 사용하면 아주 딱딱해 보일 뿐만 아니라, 화살 모양이 너무 크거나 작아서 어설퍼 보입니다.
- ✓ 화살 모양, 두께, 대시 종류, 색상 네 가지를 조정하면서 상황에 어울리는 선을 만들어냅니다.
- ✓ 선이 두 개 이상 있을 경우, 기본 선 모양을 그대로 사용하지 않고, 네 가지 요소를 조정하면서 어울리는 선을 그립니다.



## 4.2 다각형

- ✓ 연결 – 어떤 방향으로 놓여 있던 가급적 “연결”하여야 합니다. 그렇지 않으면 도형의 변화에 영향을 받습니다.
- ✓ 두께 – 외부에서 내부로 들어가면서, 또는 보다 중요한 도형으로 가면서 두께를 조절합니다.
- ✓ 농도 – 두께와 마찬가지로 선의 굵기를 조절할 수 없을 때에는 농도를 조절하여 안과 밖의 포함 관계를 표현합니다.
- ✓ 모서리 및 그림자 처리 – 모서리의 모양을 다양하게 변경하거나 그림자를 추가함으로써 변화를 줄 수 있습니다.

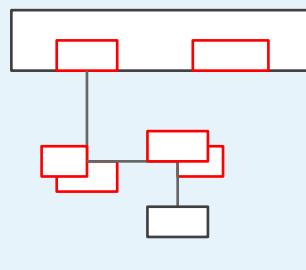
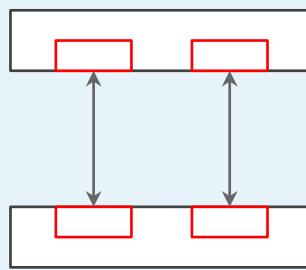
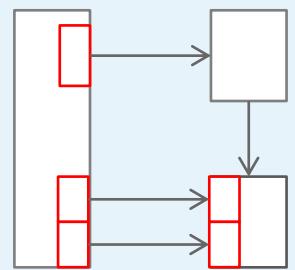


모든 다각형은 무슨 색으로든 채워야  
함. 채우지 않으면 선택이 어려움

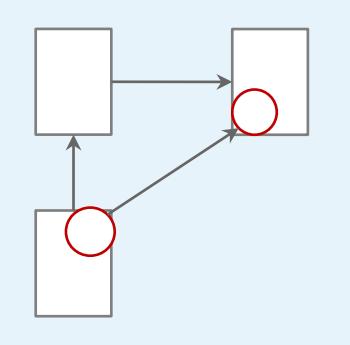
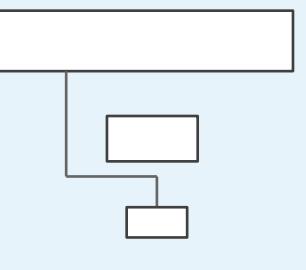
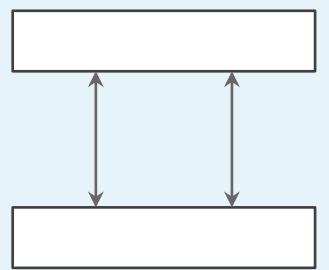
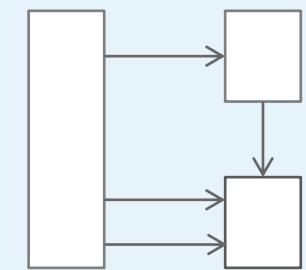
## 4.3 선 연결

- ✓ 선은 홀로 두면 안되므로 항상 연결합니다. 선 연결은 양쪽 끝에 “연결용 다각형”을 두고 연결합니다.
- ✓ 가급적이면 직선이 되도록 하며, 양쪽 끝에 두는 객체의 크기, 위치 등을 이용하여 연결을 조정합니다.
- ✓ 연결을 위한 끝 단 객체들은 그대로 남아 있거나, 그룹으로 묶인 채 남아 있습니다.
- ✓ 연결되지 않은 선은 이동이나 크기 조정 등에서 도형이 깨어질 수도 있고, 빠딱하게 연결될 수도 있습니다.

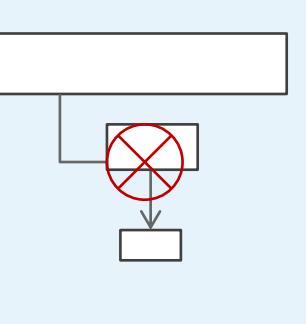
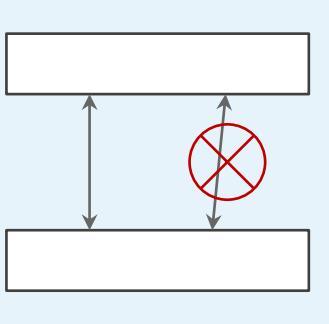
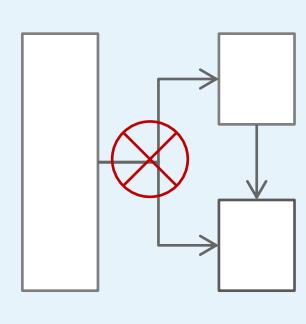
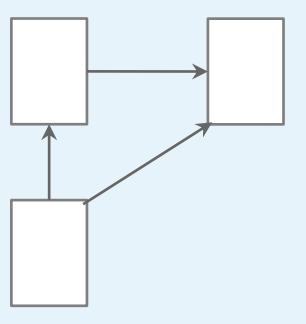
선 연결 실체



선 연결 결과 모습

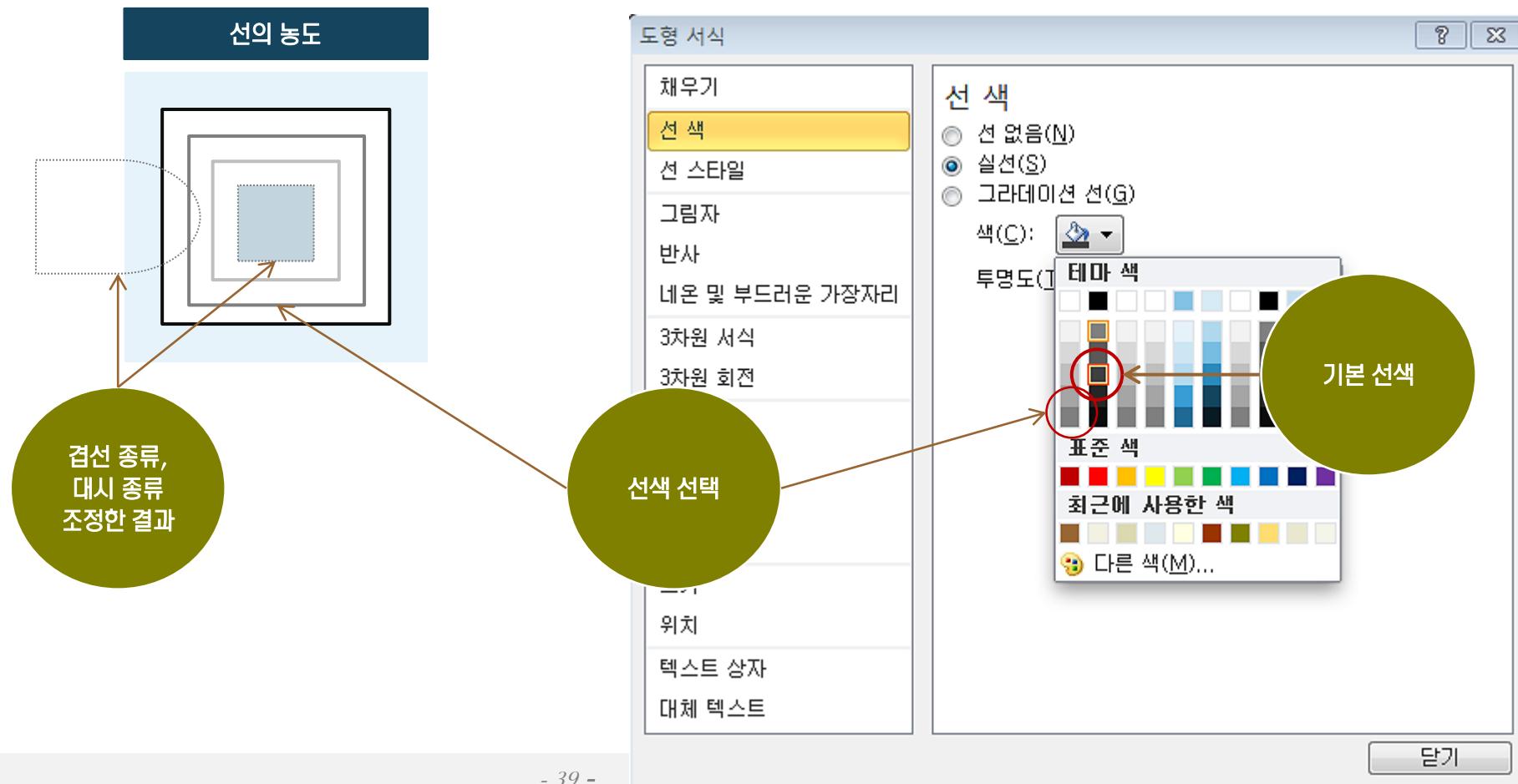


선 연결 결과 모습



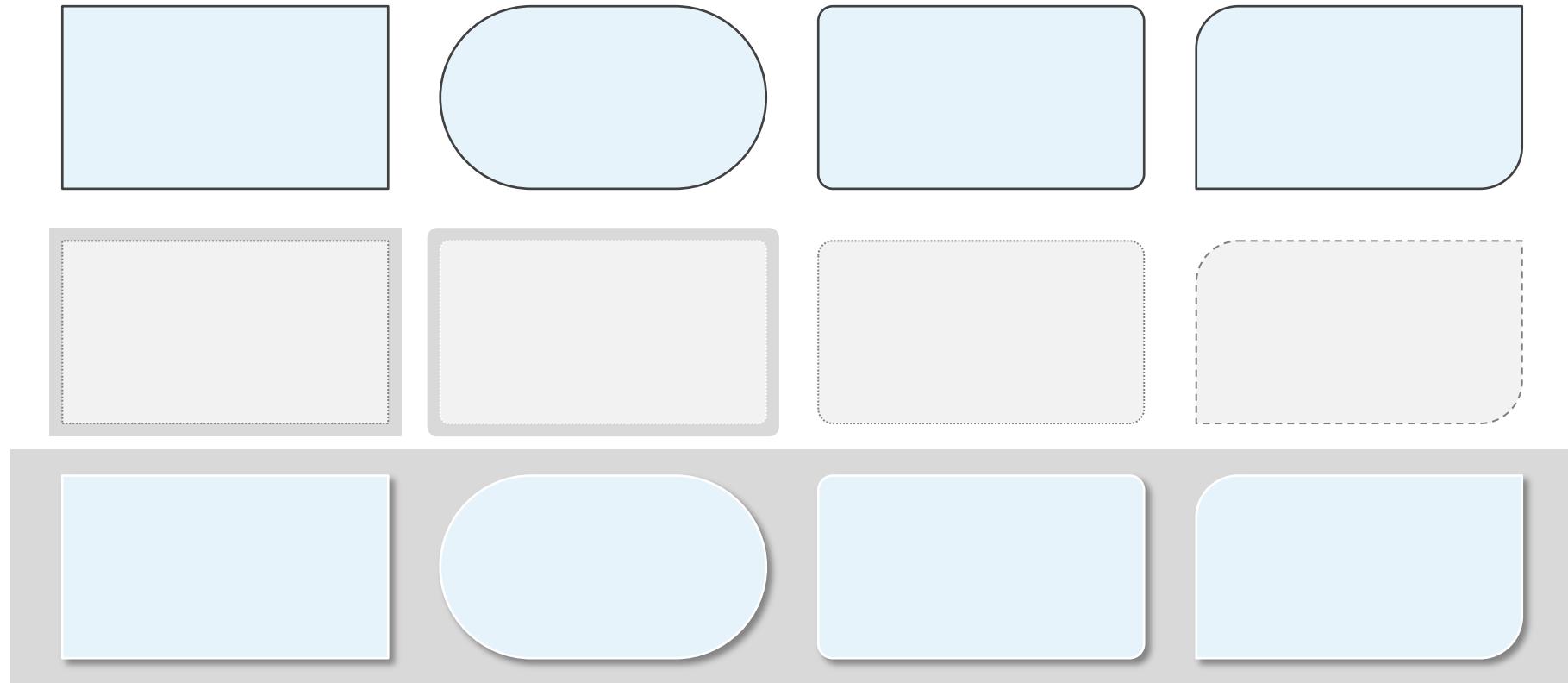
## 4.4 선 색

- ✓ 기본 선 색은 대체로 짙은 느낌입니다. 대체로 1~3 포인트 더 옅은 색을 선택합니다.
- ✓ 가장 외부 테두리를 기준으로 안쪽으로 들어갈수록 얇게 조정하면 도형이 들뜨지 않고 편하게 느껴집니다.
- ✓ 이때, 선의 두께도 미세하게 조정하면 좋습니다. 안으로 들어 갈수록 얇게 조정합니다.
- ✓ 선이 부담스러울 때는 점선으로 표현하되, “선 스타일”에서 “겹선 종류”와 “대시 종류”를 조정해서 편하게 보이도록 합니다.



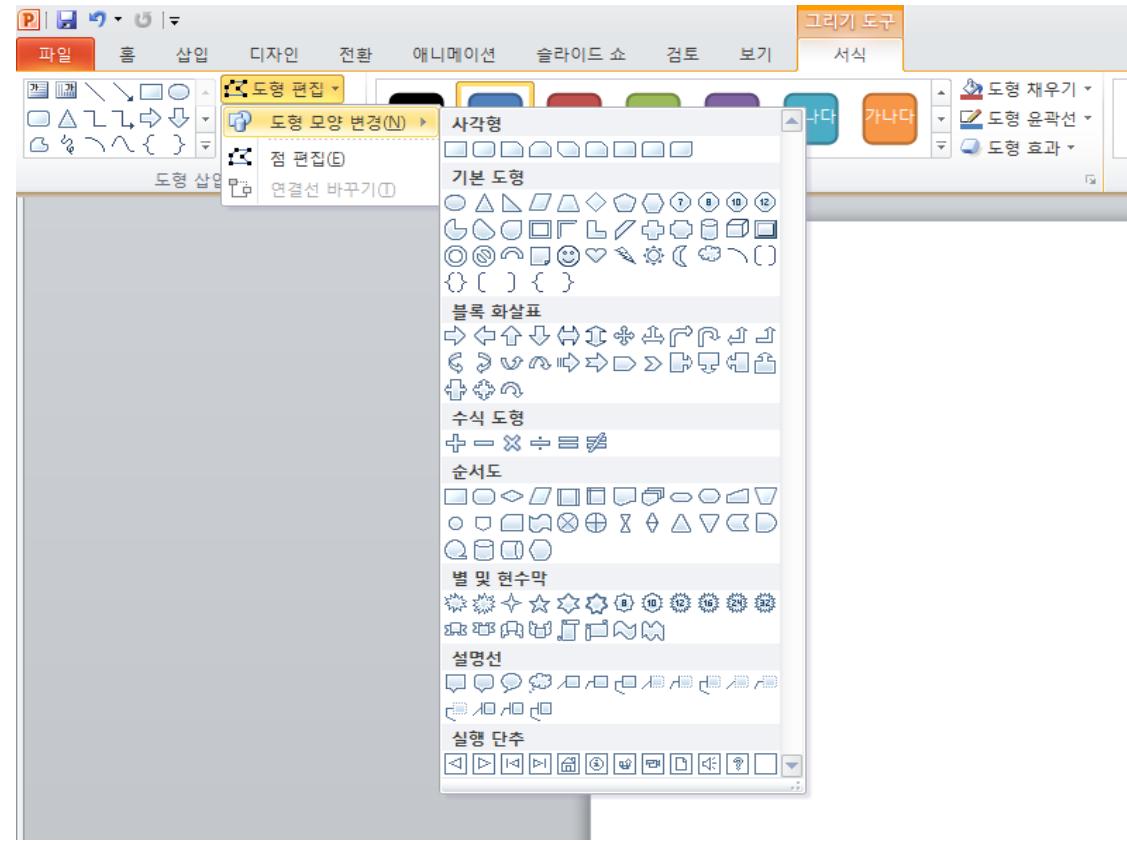
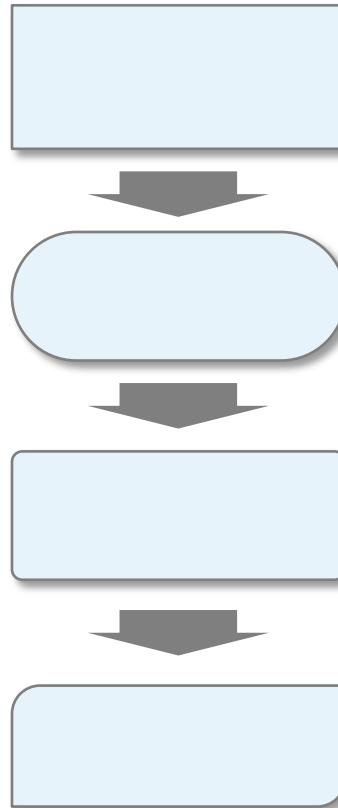
## 4.5 다각형 모양

- ✓ 도형도 선과 마찬가지로 도형의 종류, 두께, 대시 종류, 색상, 그림자 등을 조정함으로써 상황에 어울리게 그립니다.
- ✓ 다섯 가지의 조합으로 다양한 느낌을 갖는 도형을 그릴 수 있습니다.
- ✓ 때로는 배경을 사용하여 도형을 강조할 수도 있습니다. 배경색을 사용하면 흰색과 같은 밝은 색 선이 은은하게 드러납니다.
- ✓ 모서리 종류와 모서리 각의 크기를 조절함으로써 다른 느낌이 나도록 할 수 있습니다.



## 4.6 도형 변경

- ✓ 서식 메뉴에서 도형편집 → 도형모양 변경을 선택하면 다각형의 모양을 변경할 수 있습니다.
- ✓ 사각형에서 모서리가 둑근 사각형을 비롯한 다른 도형으로 모양을 변경할 수 있습니다.
- ✓ 다시 그리지 않고 변경하는 이유는, 도형에 적용된 서식을 그대로 사용하기 위해서입니다.
- ✓ 작업 중에 많이 사용되는 기능 중에 하나입니다. 처음에 선택한 다각형이 맘에 들지 않는 경우입니다.



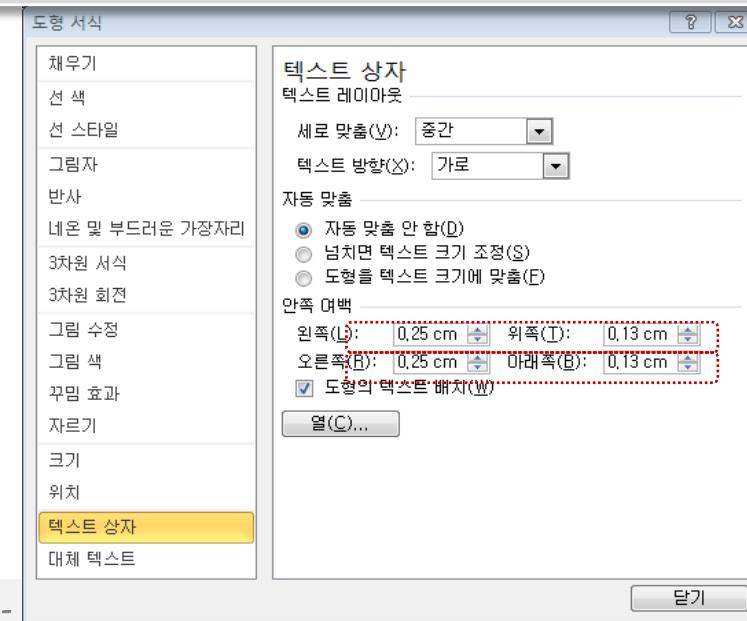
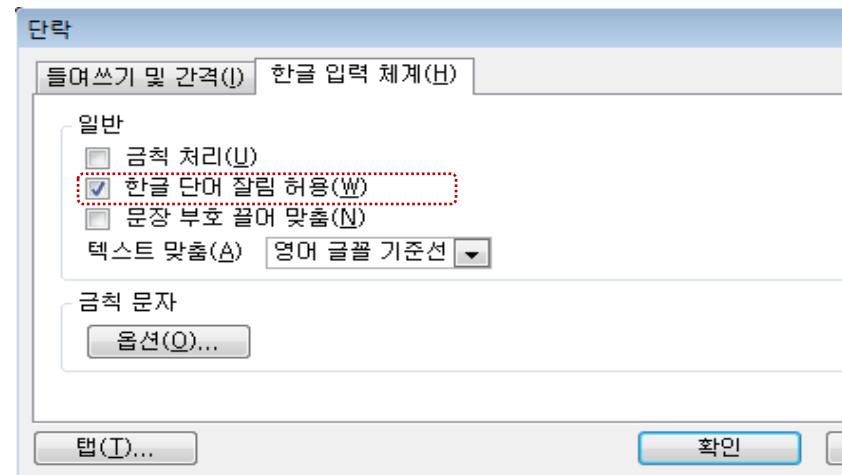
## 4.7 도표

- ✓ 프리젠테이션 도구가 제공하는 도표 기능은 제어할 수 있는 부분이 별로 없습니다. 그런 이유로 거의 사용하지 않습니다.
- ✓ 도표를 자유롭게 표현하기 위해서는 “사각형의 집합”으로 도표를 그려야 합니다.
- ✓ 복사하여 붙이기를 통해서 생각보다 쉽게 복잡한 도표를 그릴 수 있습니다. 물론 내용을 채우는 작업은 지루합니다.
- ✓ 도표를 작성자가 원하는 대로 어떤 포맷으로든 만들 수 있습니다. 뒤에 별도의 장(chapter)에서 알아 보겠습니다.



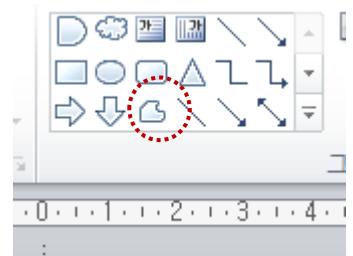
## 4.8 텍스트 상자

- ✓ 텍스트 상자는 글자를 가지런하게 들어 갈 수 있도록 단락이나 도형 서식을 조정합니다.
  - ✓ 안쪽 여백의 기본 값은 0.25cm입니다. 작은 상자에 글을 쓸 때, 이 여백은 매우 큽니다. 0.1 정도로 줄입니다.
  - ✓ 선의 두께가 두터울 때, 0.1로 하면 선에 글씨가 달라 붙어 있는 것 처럼 보입니다. 0.15나 0.2로 조정을 합니다.
  - ✓ 단어 잘림 허용을 하면, 글 상자 안에 글씨가 가득 차 보이는 효과가 있습니다. 상황에 맞게 조절합니다.



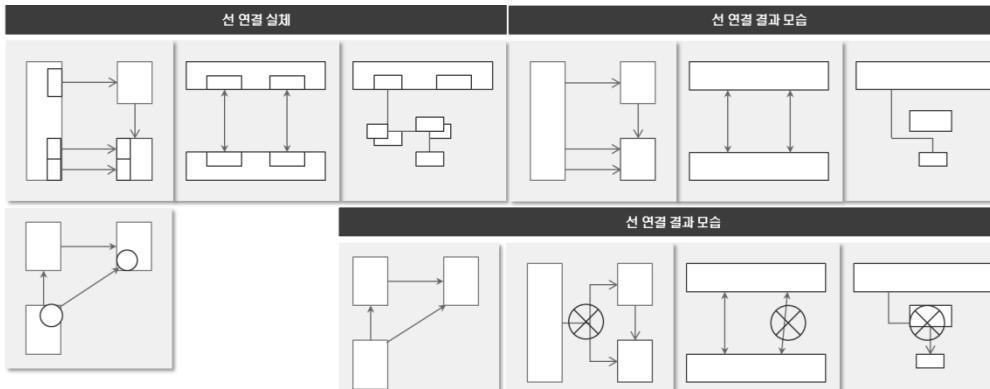
## 4.9 자유 도형

- ✓ 때로는 비정형 도형인 자유 도형을 그려야 할 때가 있습니다.
- ✓ 자유도형 아이콘을 선택한 후, 클릭으로 시작하여 그리고, 도형이 완성되는 지점에서 더블 클릭하여 마무리 합니다.
- ✓ 일부 구간에서 직선을 그릴 때, shift 키를 누르면 직선을 쉽게 그릴 수 있습니다. 그렇지 않으면 곧은 선을 그리기 힘듭니다.
- ✓ 자유 도형도 내부에 텍스트를 입력할 수 있습니다.



## 4.10 요약

- ✓ 기본 도형 그리기는 복잡한 그림으로 가는 기초가 됩니다.
- ✓ 사소한 듯 보이는 기능이 실제 표현(representation) 작업에서는 큰 효과를 가져다 줍니다.
- ✓ 예를 들면, “모든 선은 연결한다.”라는 원칙을 기반으로 연결 선으로 그리면 팀 작업이나 확장 작업에 도움이 됩니다.
- ✓ 기본 도형을 그대로 사용하는 것이 아니라, 항상 상황에 맞도록 조정하여 사용하는 습관을 길러야 합니다.



차세대 000 프		3월	4월	5월	6월	7월	8월
인프라 팀	주요활동	릴리즈 1				릴리즈 2	
	개발자	6	6	6	6	6	6
000 팀	주요활동	서비스 제어/채널 인프라 릴리즈		베타 버전 릴리즈			
	개발자	0	1/4	1/4	1/4	1/4	1/4
PM 그룹	주요활동	PMO		PMO			
	관리자	2/2	2/2	2/2	2/2	2/2	2/2
인력 합계	내부인력	2	6	6	6	6	6
	외부인력	8	8	8	8	8	8



## 5. 실습 #2 – 레이아웃

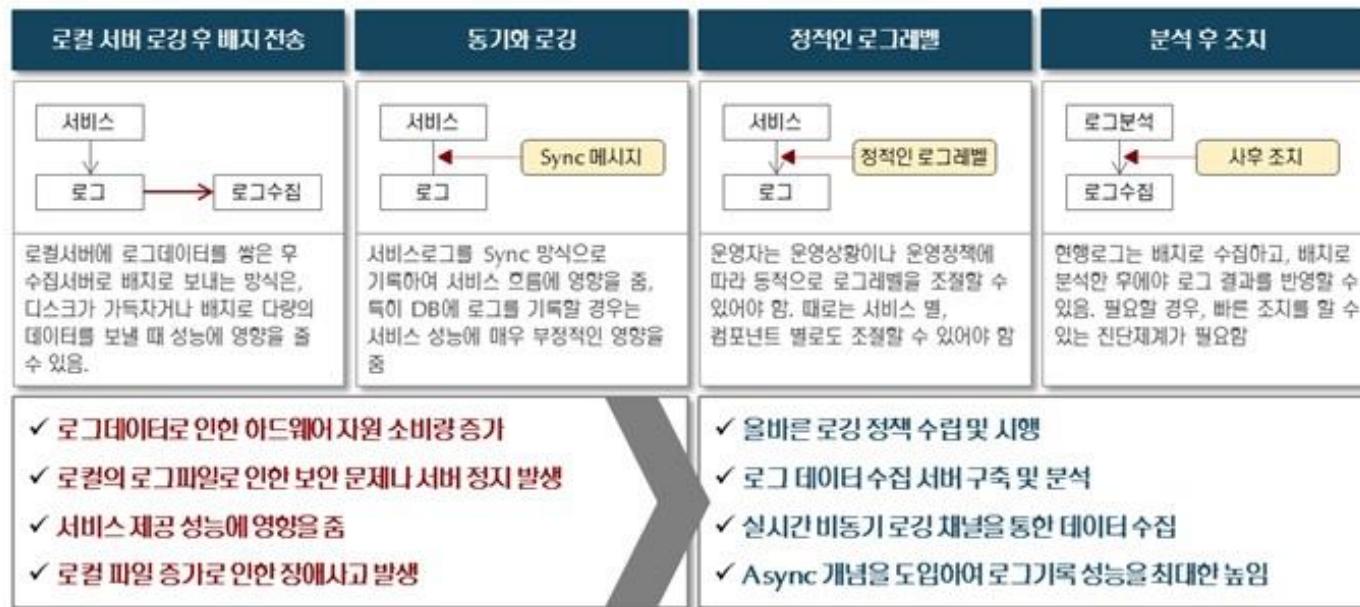
- 
- 5.1 레이아웃 실습 - 스마트 로깅
  - 5.2 그리기 안내 1 – 레이아웃
  - 5.3 그리기 안내 2 – 설명 영역 나누기
  - 5.4 그리기 안내 3 – 작은 도형 그리기
  - 5.5 그리기 안내 4 – 일취월장
  - 5.6 그리기 안내 5 – 세부 절차
  - 5.7 그리기 안내 6 – 레이어 개념
  - 5.8 그리기 안내 7 – 연결선 조정
  - 5.9 요약

# 5.1 레이아웃 실습 – 스마트 로깅

- ✓ 예시는 상단 텍스트 영역과 하단 다이어그램 및 설명 영역으로 나누었습니다. 그대로 따라 그려 봅니다.

## 2. 스마트로깅 개요

- ✓ 로깅 구조는 수행성능, 스토리지 자원, 보안, 장애, 운영분석 등에 영향을 주는 매우 중요한 아키텍처 요소임
- ✓ 현행 로깅 구조는 성능, 자원, 보안, 분석 등의 측면에서 스마트하지 못한 구조를 가지고 있음
- ✓ 올바른 로깅 구조 설계 및 기술적용을 통해 스마트 로깅 목표를 달성하여야 함



올바른 성장과 따뜻한 나눔...

| -4 -

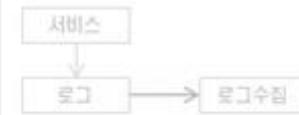
## 5.2 그리기 안내1 – 레이아웃

- ✓ 텍스트 영역과 그림 영역을 나누고, 전체 영역을 중앙에서 반으로 나누었습니다.

### 2. 스마트로깅 개요

- ✓ 로깅 구조는 수행성능, 스토리지 자원, 보안, 장애 회피 등에 주는 매우 중요한 아키텍처 요소임
- ✓ 현행 로깅 구조는 성능, 자원, 보안, 분석 등의 능력에서 스마트로깅 구조를 가지고 있음
- ✓ 올바른 로깅 구조 설계 및 기술적용을 통해 스마트 로깅 목표를 달성하여야 함

#### 로컬 서버 로깅 후 배치 전송



로컬서버에 로그데이터를 성은 후 수집서버로 배치로 보내는 방식은, 디스크가 가득차거나 빠자로 대량의 데이터를 보낼 때 성능에 영향을 줄 수 있음.

- ✓ 로그데이터로 인한 하드웨어 자원 소비량 증가
- ✓ 로컬의 로그파일로 인한 보안 문제나 서버 정지 발생
- ✓ 서비스 제공 성능에 영향을 줌
- ✓ 로컬 파일 증가로 인한 장애사고 발생

#### 동기화 로깅



서비스로그를 Sync 방식으로 기록하여 서비스 흐름에 영향을 줌. 특히 DB에 로그를 기록할 경우는 서비스 성능에 매우 부정적인 영향을 줌

- ✓ 올바른 로깅 정책 수립 및 시행
- ✓ 로그 데이터 수집 서버 구축 및 분석
- ✓ 실시간 비동기 로깅 채널을 통한 데이터 수집
- ✓ Async 개념을 도입하여 로그 기록 성능을 최대한 높임

#### 정적인 로그레벨



운영자는 운영상황이나 운영정책에 따라 동적으로 로그레벨을 조절할 수 있어야 함. 때로는 서비스 레벨, 커뮤니티 레벨로 조절할 수 있어야 함

#### 분석 후 조치

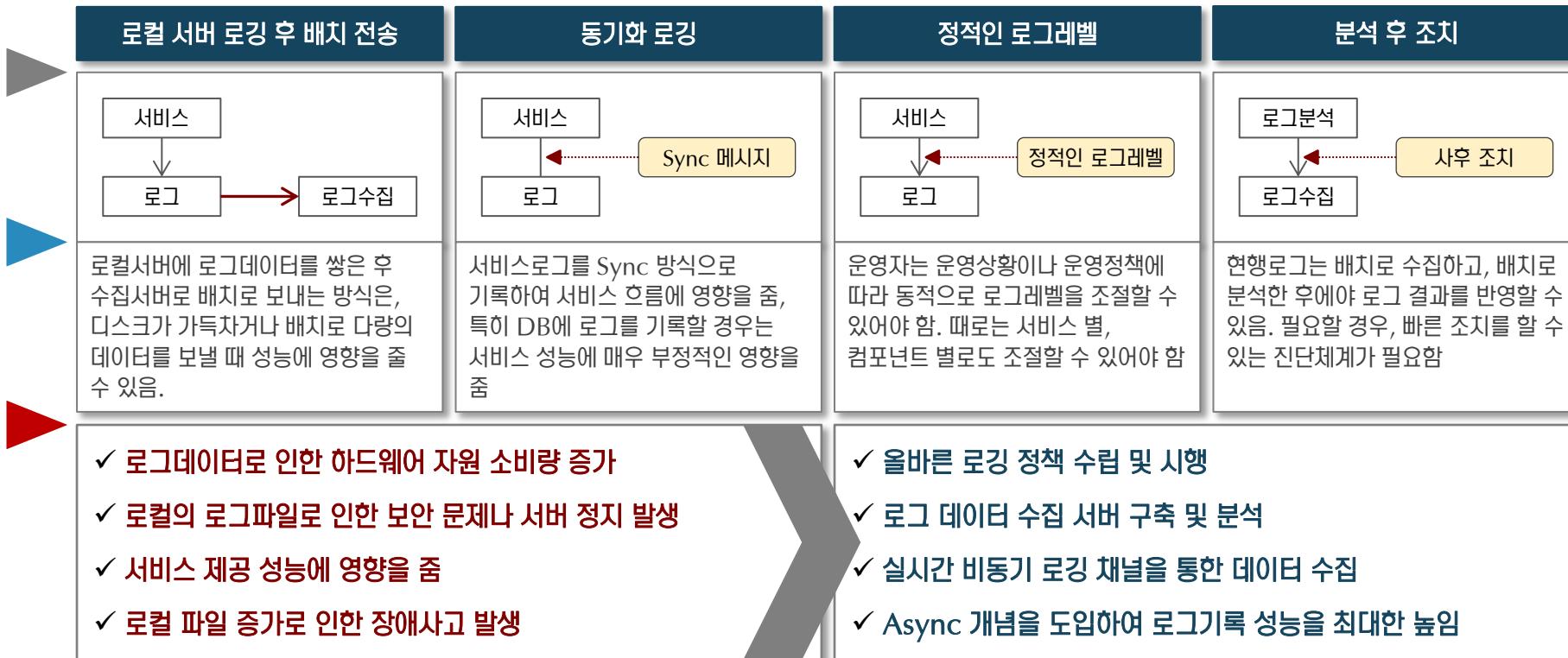


인행로그는 배치로 수집하고, 배치로 분석한 후에야 로그 결과를 반영할 수 있음. 필요할 경우, 빠른 조치를 할 수 있는 전단계가 필요함

올바른 성장과 따뜻한 나눔...

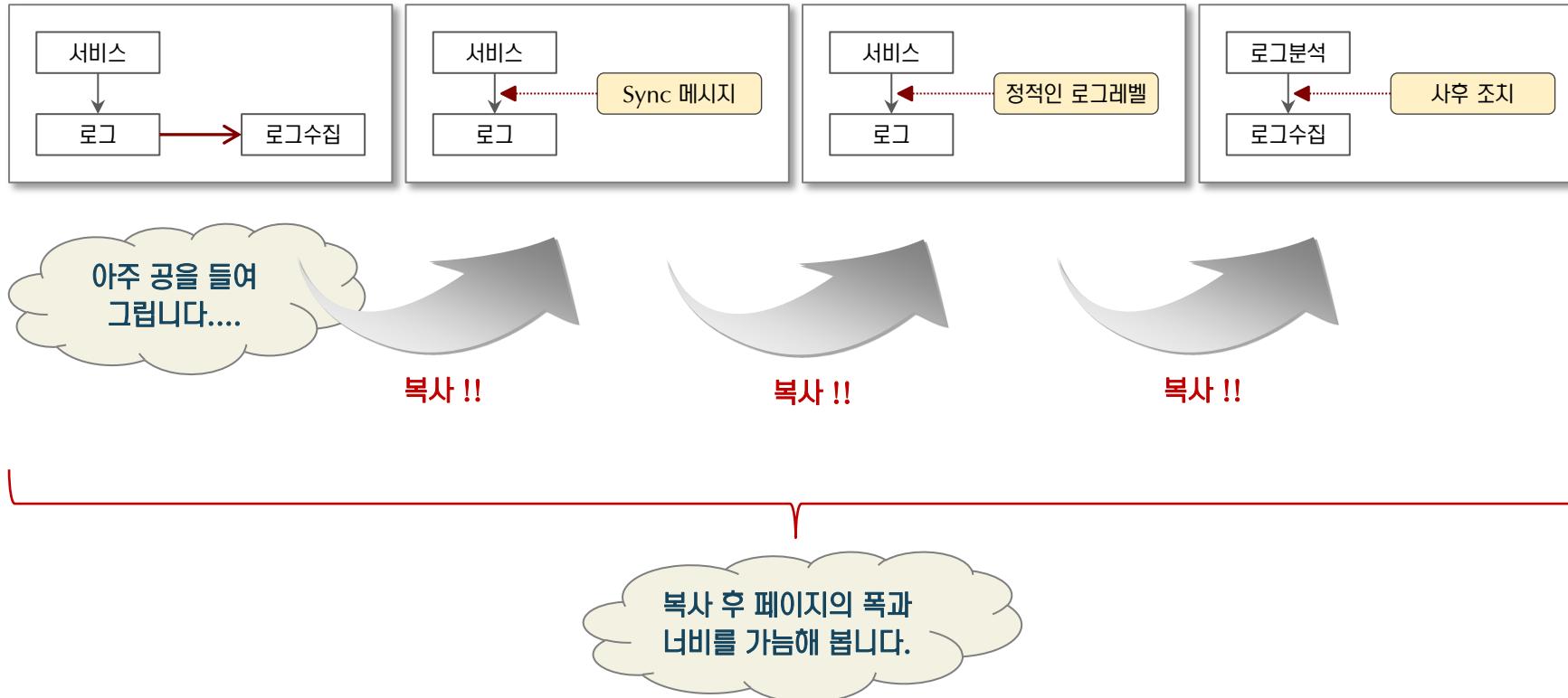
## 5.3 그리기 안내 2 – 설명 영역 나누기

- ✓ 설명 영역을 문제를 서술하는 영역과 개선 목표 영역으로 나누었습니다.
- ✓ 문제 서술 영역은 다시 그림 영역과 문제를 서술하는 영역으로 나누었습니다.
- ✓ 상단에 각 영역 별로 문제를 설명하는 타이틀을 추가했습니다.
- ✓ 한 페이지에서 무엇을 하고자 하는지, 명확하게 설명하고 있습니다.



## 5.4 그리기 안내 3 – 작은 도형 그리기

- ✓ 그림으로 상황을 간결하게 묘사함으로써, 설명이 부족한 부분을 보충하고 있습니다.
- ✓ 간결하고 작은 그림은 몇 줄의 설명보다 더 큰 전달 효과를 보여 줍니다.
- ✓ 지루한 텍스트로 가득 찬 페이지에 간결한 그림을 추가함으로써 문제에 대한 가시성을 높입니다.



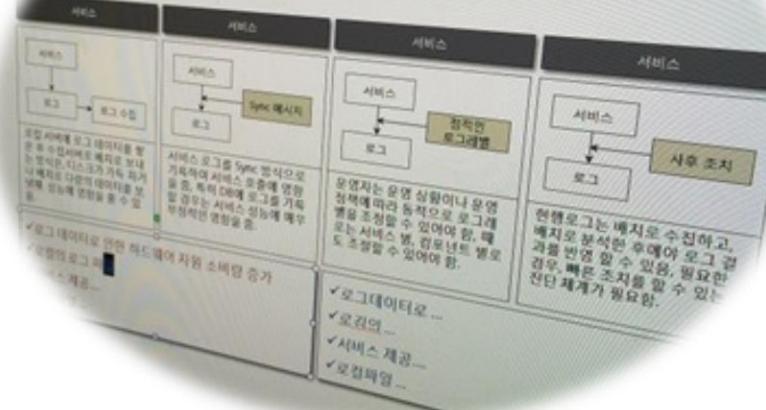
## 5.5 그리기 안내 4 – 일취월장

- ✓ 초보자가 기본 설정값을 이용하여 도형을 그리는 것은 감당하기 힘들 정도로 좋은 도구를 손에 들고 있는 것과 같습니다.
  - ✓ 특히, 엔지니어가 작성하는 발표자료는 설계 문서나 의사결정을 위한 기초 자료로 사용되는 경우가 많습니다.
  - ✓ “사용자 정의 도형”을 바탕으로 기본 세트를 만들고, 기본 세트를 확장하여 가는 방식으로 자료를 만들어야 합니다.
  - ✓ 그러면, 도형을 그리는 작업이 편안하다는 느낌을 처음 갖게 됩니다. 그런 상태에서 몇 가지를 살펴봅니다.

## 프리젠테이션 도구가 제공하는 설정값을 가진 기본 도형

## 기본도형(16pt)

가나다 가나다 가나다 가나다 가나다 가나다  
그리고 대라마 대라마 대라마 대라마 대라마 대라마 대라마  
네시야 네시야 네시야 네시야  
자카 자카 자카 자카 자카 자카 자카

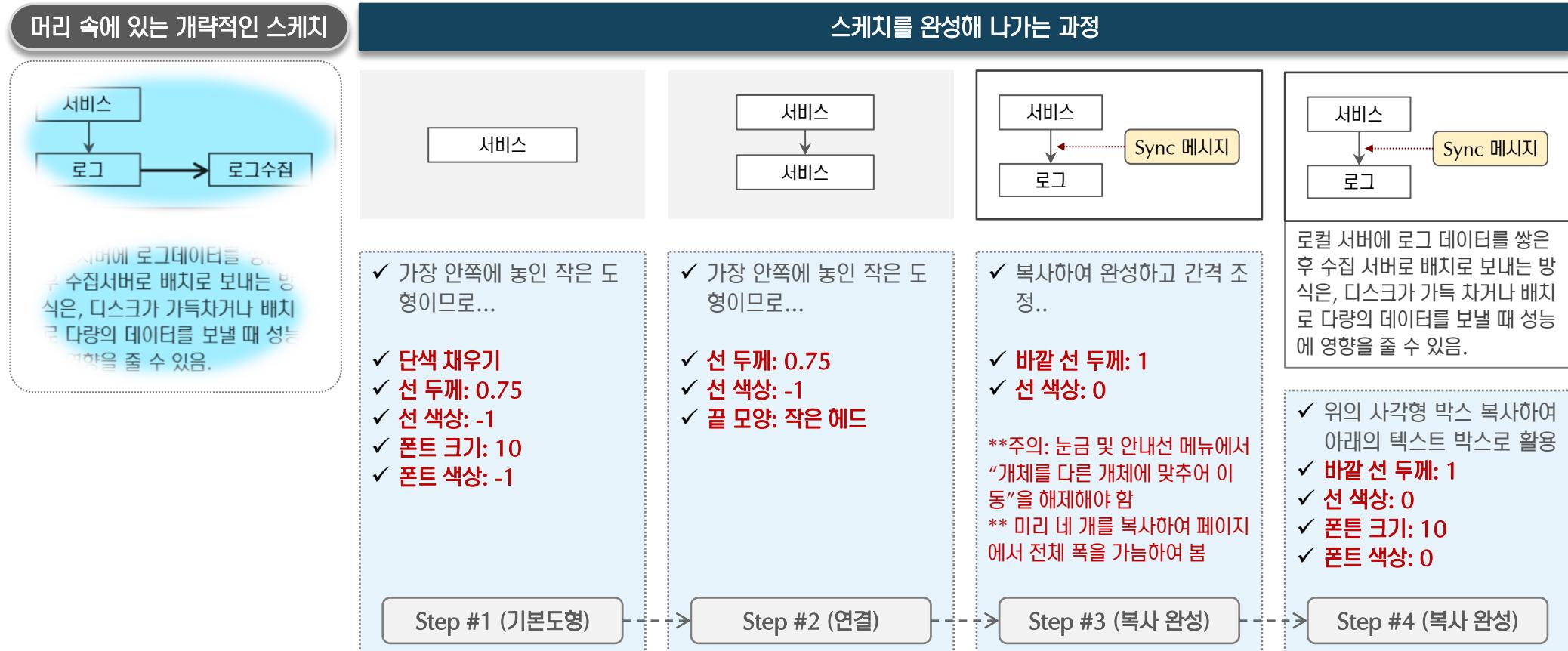


## 사용자가 정의한 사용자 정의 도형과 선

기본도형(12pt) → 기본도형(12pt)

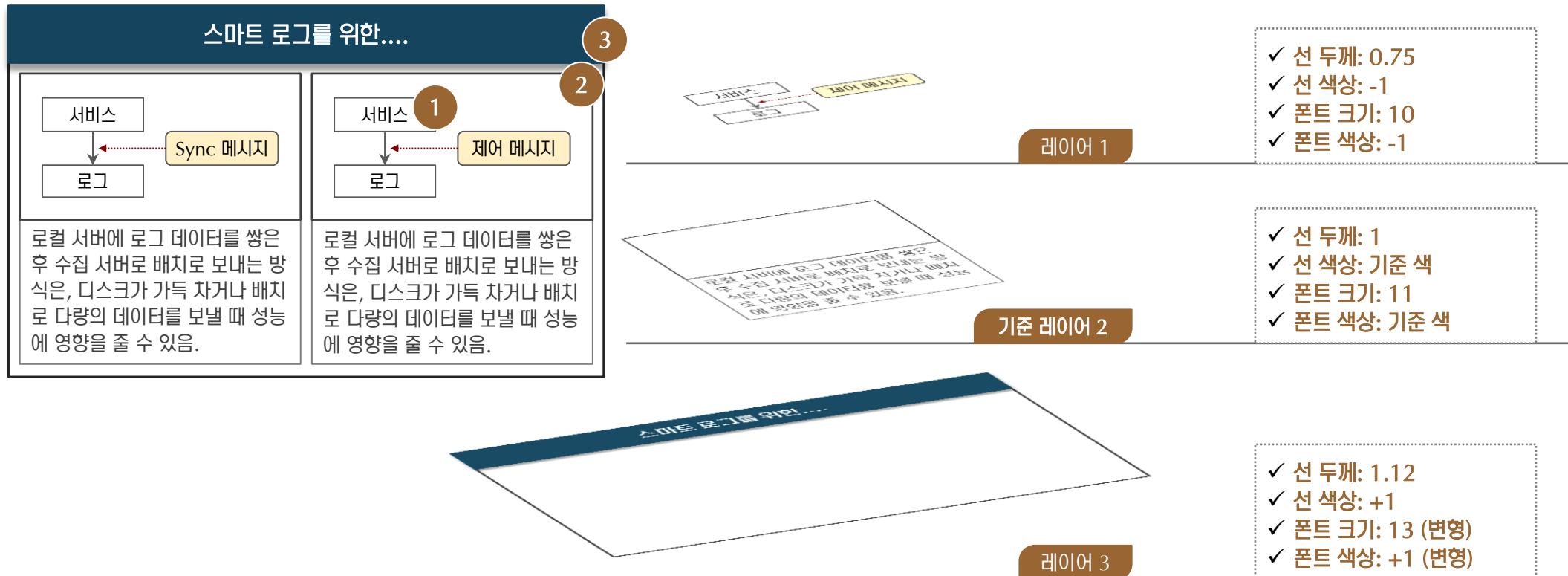
## 5.6 그리기 안내 5 – 세부 절차

- ✓ 레이아웃은 머리 속에 개략적으로 그려 두고, 실제 작업은 기본 세트에 집중합니다.
- ✓ 기본 세트를 완성할 때, 항상 첫 번째 도형(여기서는 사각형)에 충분한 시간을 들입니다.
- ✓ 첫 번째 도형을 두고 선 색, 두께, 채우기, 폰트 크기, 폰트 색상을 조절해 가면서 마음에 들 때까지 조정합니다.
- ✓ 도형을 완성하고, 텍스트 박스는 그 도형에서 사용한 박스를 그대로 내려서 사용합니다.



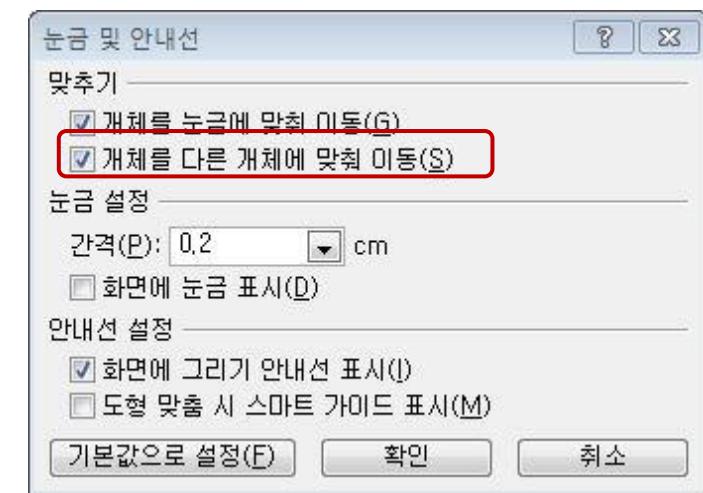
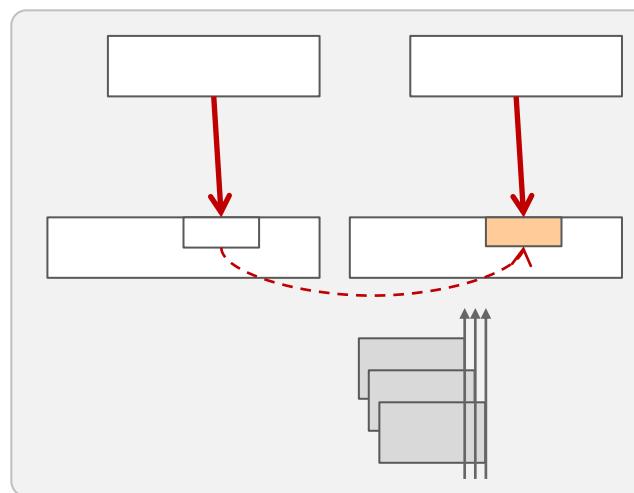
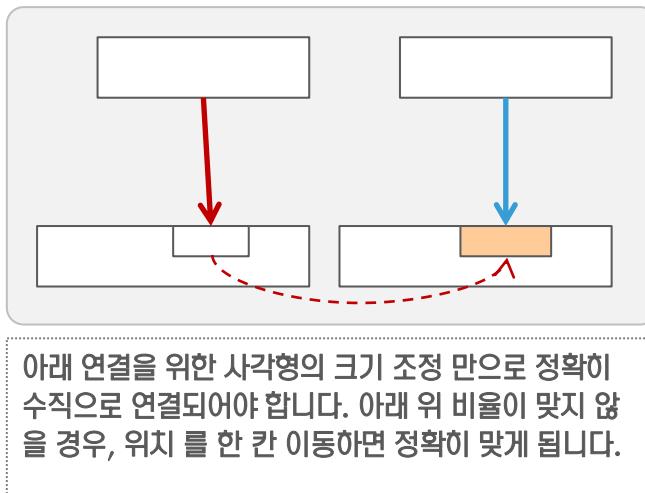
## 5.7 그리기 안내 6 – 레이어 개념

- ✓ 레이아웃은 머리 속에 개략적으로 그려 두고, 실제 작업은 기본 세트에 집중합니다.
- ✓ 기본 세트를 완성할 때, 항상 첫 번째 도형(여기서는 사각형)에 충분한 시간을 들입니다.
- ✓ 첫 번째 도형을 두고 선 색, 두께, 채우기, 폰트 크기, 폰트 색상을 조절해 가면서 마음에 들 때까지 조정합니다.
- ✓ 도형을 완성하고, 텍스트 박스는 그 도형에서 사용한 박스를 그대로 내려서 사용합니다.



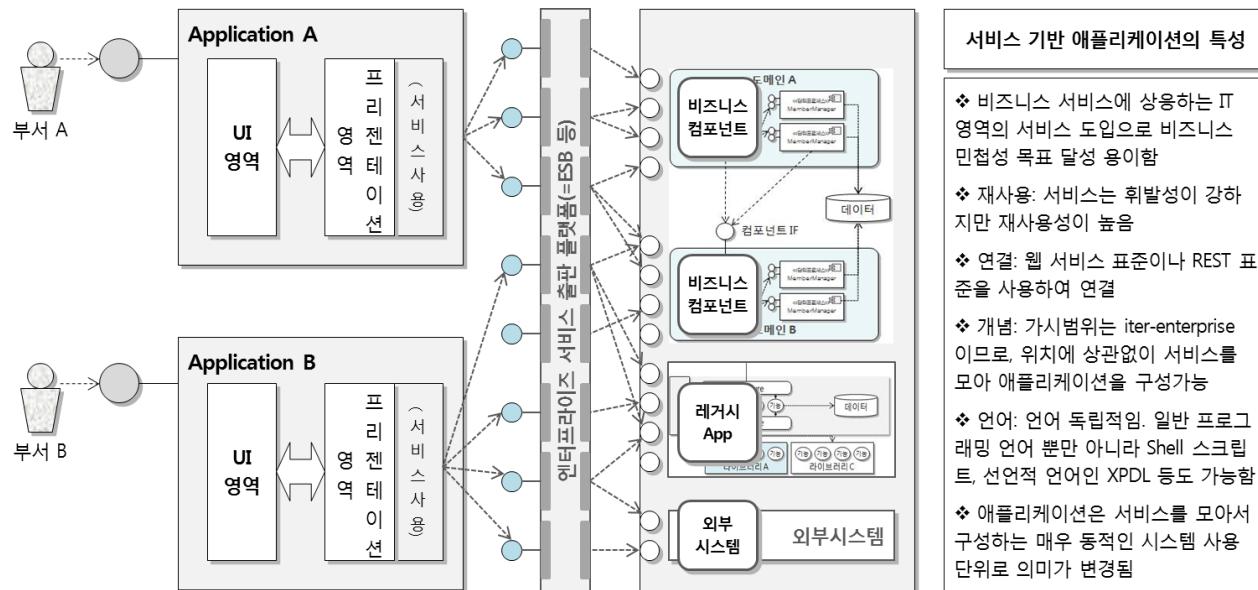
## 5.8 그리기 안내 7 – 연결선 조정

- ✓ 컨트롤 키를 이용한 미세 조정을 하지 않았음에도 직선이 곧게 연결되지 않는 경우가 있습니다.
- ✓ 조절 간격이 제 멋대로 움직이는 듯한 기분이며, 작업자는 정확히 연결하기 위해서 미세 조정을 하게 됩니다.
- ✓ 실제로 지정된 눈금(여기서는 0.2 cm) 단위로 움직이지 않습니다.
- ✓ 이런 현상의 비밀은 바로 눈금 및 안내서 메뉴에서 “개체를 다른 개체에 맞춰 이동”을 선택했기 때문입니다.



## 5.9 요약

- ✓ 레이아웃은 최대한 단순하게 가져가며, 전 페이지에 걸쳐서 세 가지 유형 이하로 제한합니다.
- ✓ 레이아웃 속의 레이아웃(세부 구도)은 페이지의 내용을 지속적으로 분해하면서, 각 계층에서 균형을 갖도록 합니다.
- ✓ 페이지를 처음 보았을 때, 구도가 한 눈에 파악이 되지 않으면, 레이아웃 설정이나 작업에 문제가 있다는 증거입니다.
- ✓ 레이아웃은 전체 발표 자료가 일관된 모습으로 보이도록 해주는 틀입니다. 모든 컨텐츠 사이로 보이지 않는 선이 있습니다.





## 6. 실습 #3 – 도표

- 
- 6.1 도표 실습
  - 6.2 그리기 안내 1 – 표 기능
  - 6.3 그리기 안내 2 – 기본 세트 그리기
  - 6.4 그리기 안내 3 – 크기 조정
  - 6.5 표 응용 예제: 프로젝트 수행계획
  - 6.6 세밀한 도표
  - 6.7 요약

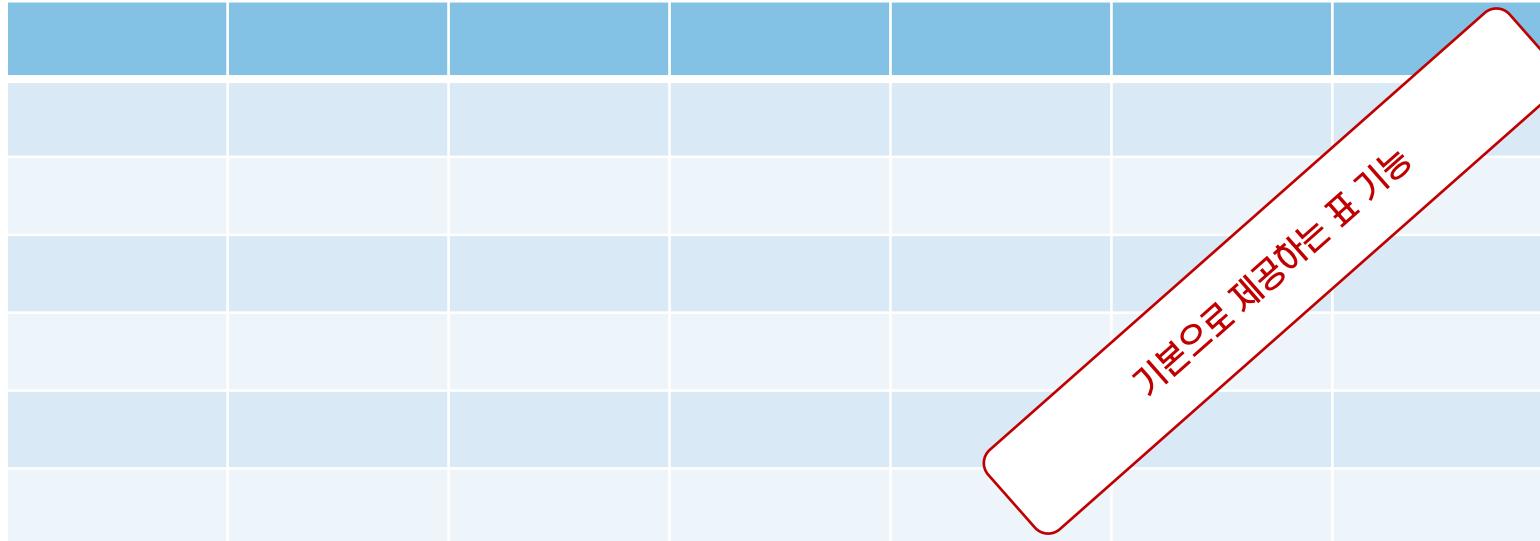
# 6.1 도표 실습

- ✓ 프리젠테이션 도구가 제공하는 도표(테이블) 기능은 제약 조건이 많습니다.
- ✓ 따라서, 사각형의 끝없는 복사하여 붙이기 기능을 이용하여 도표를 만듭니다.
- ✓ 어떤 형태의 변형이라도 모두 수용할 수 있으므로, 많이 사용됩니다.
- ✓ 힘들어 보이지만, 결국은 기준 세트를 만들고, “복사하여 붙이기” 기능을 이용하여 만들어 가므로, 생각보다 쉽습니다.

차세대 000 프로젝트 수행계획												
인력 합계	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월	인력 합계	
	준비	릴리즈 1				릴리즈 2				릴리즈 3 – 서비스 준비		
	반복 1	반복 1	반복 2	반복 3	반복 4	반복 1	반복 2	반복 3	반복 4	반복 1	반복 2	시범 운영
	인프라 팀	서비스 제어/채널 인프라 릴리즈				아키텍처 이슈 해결 및 지원				통합 및 테스트	시범 운영	
000 팀	주요활동	서비스 제어/채널 인프라 릴리즈				아키텍처 이슈 해결 및 지원				통합 및 테스트	시범 운영	
	개발자	6	6	6	6	6	6	6	6	6	4	58
PM 그룹	주요활동	베타 버전 릴리즈				RSD 릴리즈				통합 및 테스트	시범 운영	
	개발자	0	1/4	1/4	1/4	1/6	1/6	1/6	1/6	1/4	1/4	42
인력 합계	주요활동	PMO				PMO				PMO		
	관리자	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	2/2	20
인력 합계	내부인력	2	6	6	6	6	6	6	6	6	6	56
	외부인력	8	24	24	24	30	30	30	22	22	20	234

## 6.2 그리기 안내 1 – 표 기능

- ✓ 기본으로 제공하는 “표 기능”은 간단한 표를 그릴 때는 부담 없이 사용할 수 있습니다.
- ✓ 하지만, 표가 복잡하고, 선, 폰트, 셀 등을 자유롭게 조정해야 하는 상황에서는 사용하기 어렵습니다.
- ✓ 엑셀로부터 “개체 삽입”을 통해서 표를 그릴 때는 계산 등은 편리하지만, 보기 좋은 도표를 그리는 것이 불가능합니다.
- ✓ 발표 자료의 표현에서 표가 필요할 경우, 사각형을 직접 쌓아서 그리는 것을 추천합니다.



## 6.3 그리기 안내 2 – 기본 세트 그리기

- ✓ 실습용 예제 도표의 핵심은 월-릴리스-반복입니다. 이것을 기본 세트로 보고 그리기를 시작합니다.
- ✓ 기본 세트를 그리는데 시간을 충분하게 들입니다. 크기, 색상, 선, 폰트, 배경색 등을 세밀하게 조정합니다.
- ✓ 기본 세트에 대한 작업이 끝나면 크기가 적절한지 “복사하여 붙이기” 기능을 이용하여 확인합니다.

그리기 순서 1

- ✓ 사각형을 그리고
- ✓ 크기를 조정하고,
- ✓ 채우기 색을 조정하고,
- ✓ 폰트 크기와 색상을 조정하고,
- ✓ 선 색상과 두께를 조정하고,
- ✓ 복사하여 붙이기 반복...

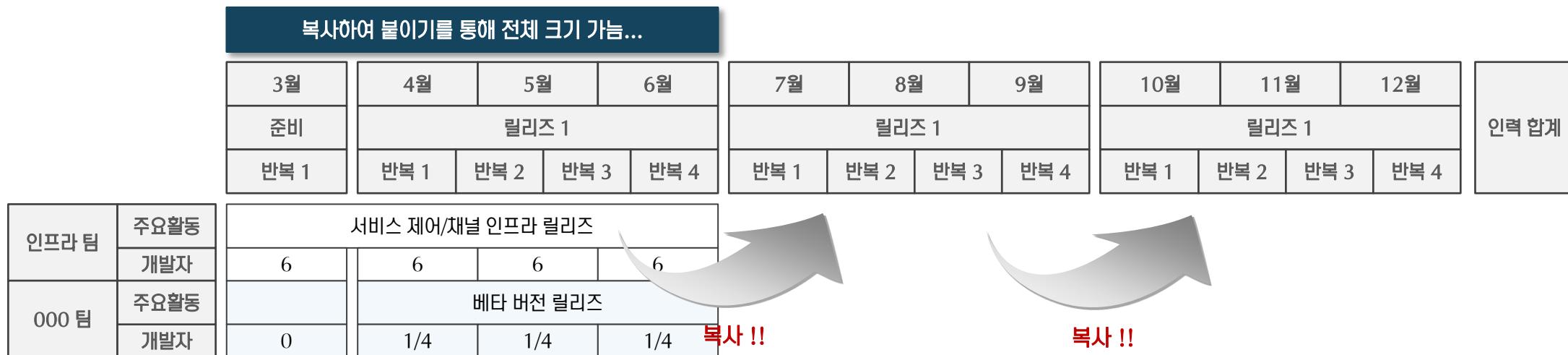
기본 세트 작업...

3월	4월	5월	6월
준비	릴리즈 1		
반복 1	반복 2	반복 3	반복 4

기본 세트를 만드는데 많은 시간을 들입니다.!!

## 6.4 그리기 안내 3 – 크기 조정

- ✓ 복사하여 붙이기를 하여, 크기가 적절한지 판단합니다.
- ✓ 원하는 정보를 다 담을 수 없다고 판단하면, 기본 세트의 크기를 조절합니다. 이 때, 마찬가지로 충분히 시간을 들입니다.
- ✓ 오른쪽으로 복사하여 확인한 후, 아래쪽으로 복사하여, 마찬가지로 충분한 공간이 있는지 확인합니다.
- ✓ 이 과정이 모두 끝난 후, 기본 세트의 크기가 적절하다고 판단하면, 한 차례 더 조정 과정을 거칩니다.



## 6.5 표 응용 예제: 프로젝트 수행계획

- ✓ 현장에서 사용하였던 **프로젝트 수행 계획 예제**를 통해서 “자유롭게 표를 그릴 수 있음”을 보여 줍니다.
- ✓ 프로젝트 기간을 16개월로 설정하고 프로젝트 수행 계획을 작성하였습니다.
- ✓ 인프라, 운영 등도 아래의 틀 속에서 큰 일정 계획을 수립할 수 있습니다.
- ✓ 차세대 시스템 구축은 업무 간의 의존성 분석을 기반으로 “00 시스템”으로부터 시작합니다.

연도	2013년								2014년								2015년														
릴리스	7월	8월	9월	10월	11월	12월	1월	2월	3월	릴리스 1(알파1)	릴리스 2(알파2)	릴리스 3(베타1)	릴리스 4(베타2)	릴리스 5(RC1)	릴리스 6(ver 1.0)																
월	7월	8월	9월	10월	11월	12월	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월	1월	2월	3월	4월	5월	6월	7월	8월	9월				
주요 이벤트	파일럿시작								파일럿종료								인력투입								릴리스 시작						
월	7월	8월	9월	10월	11월	12월	1월	2월	3월	4월	5월	6월	7월	8월	9월	10월	11월	12월	1월	2월	3월	4월	5월	6월	7월	8월	9월				
구분	프로젝트준비								프로젝트수행																9월						
마일스톤	기획	파일럿		수행업체선정	팀워크	대출시스템		채권,영업		서비스,콜센터, Hpg		회계, 통계		일관화																	
기술트랙	→ 아키텍처설계 및 시행				대외인터페이스, 보안				배치 아키텍처				경험 자원화																		
비즈트랙	BPR		비즈니스 프로세스 지속적인 피드백 및 반영(목표: BPR 결과 모델과 개발 내역 동기화)																												
변화관리	운영팀 보완	기술교육 기본		기술교육 심화		의사결정 및 보고		○ ○ ○ 지속적인 변화관리		표를 마음대로 그릴 수 있음을 보여 줍니다.											✓ 문서와 모델	✓ 프레임워크	✓ 라이브러리	✓ 교육콘텐츠 등							
인프라	● ● ●	표를 마음대로 그릴 수 있음을 보여 줍니다.											표를 마음대로 그릴 수 있음을 보여 줍니다.																		
표를 마음대로 그릴 수 있음을 보여 줍니다.																															

## 6.6 세밀한 도표

- ✓ 도형을 이용하여 그릴 때, 폰트, 간격 조정 등이 자유롭습니다. 특히 복잡한 도표를 그릴 때 매우 유용합니다.
- ✓ 아래와 같이 서로 다른 크기의 도표, 그리고 추가 설명용 텍스트 박스 등은 표준 도표를 이용하여 그리기 어렵습니다.
- ✓ 복잡한 듯 보이지만, 이 역시 복사하여 붙여넣기를 통해 생각보다 빨리 작성할 수 있습니다.

SW 엔지니어링 지식 다지기 (8주)		
주차	넓이 기술 (집합교육, 3h/일)	깊이 기술 (현장교육, 5h/일)
1	1주차 Java	Java - 주제 1: 발표 2회 - 주제 2: 발표 1회 - 주제 3: 글쓰기
2	2주차 UML/객체모델링	
3	3주차 HTML/CSS	
4	4주차 JSP/Servlet	
5	5주차 Javascript	Javascript - 주제 1: 발표 2회 - 주제 2: 발표 1회 - 주제 3: 글쓰기
6	6주차 RDBMS	
7	7주차 SWT	
8	8주차 Unit test/영상관리	
기술의 넓이를 추구하는 집합교육은 일일 3시간, 오전에 수행함 기술의 깊이를 추구하는 현장학습은 일일 5시간, 오후에 수행함 집합교육과 현장 학습은 월~목요일까지 하며, 금요일은 오전은 발표와 글쓰기를 하고, 오후는 아웃도어 활동		

SW 엔지니어링 실무역량 갖추기 (16주)			
주차	이론 보완/실무 가이드 (집합교육, 3h/일)	실습 프로젝트 (현장실습, 5h/일)	
1	9주차 Java/UML	실습 시스템 결정	아카 1.1.1
2	10주차 시스템 분석	요구사항 명세/분석/개발	
3	11주차 Java IO, SWT	파일, GUI로 개발	아카 2.1.2
4	12주차		
5	13주차 HTML/CSS	웹 UI로 개발	아카 3.1.2
6	14주차 JSP/Servlet		
7	15주차 Javascript		
8	16주차 RDBMS, DB 모델링	RDB로 데이터 저장 개발	아카 3.1.3
9	17주차		
10	18주차 Spring DI/MVC	웹앱 프레임워크 적용 개발	아카 4.1.3
11	19주차		
12	20주차 myBATIS	데이터 프레임워크 적용 개발	아카 4.1.4
13	21주차 JPA	JPA 적용 개발	아카 4.1.5
14	22주차 비즈니스 컴포넌트	비즈니스 컴포넌트 적용 개발	아카 4.2.5
15	23주차 도메인 모델 업데이트	업무 이해 및 기능 추가	아카 4.2.5
16	24주차 프로젝트 마무리, 문서작업, 프로젝트 완료 보고		

소통 역량 다지기 (3주)		
주차	넓이 기술 (집합교육, 3h/일)	깊이 기술 (현장교육, 5h/일)
1	25주차 프리젠테이션 교육	발표 자료 준비
2	26주차 글쓰기 교육	글쓰기
3	27주차 인큐베이션 수료 프리젠테이션	

오전 3시간은 프리젠테이션과 글쓰기 교육을 수행함  
 오후 5시간은 프리젠테이션 준비와 글쓰기를 함  
 1,2주차 금요일 오전은 발표를 하고, 오후는 아웃도어 활동  
 인큐베이션 수료 프리젠테이션은 모든 수료생이 수행함

본 과정은 웹기반 비즈니스 시스템 개발을 주로 하는  
 엔지니어에 맞추었습니다. 수료 후 주니어 웹 프론트  
 엔지니어 자격증이나 주니어 서버 엔지니어  
 자격증을 획득할 수 있습니다.

Java 기반의 범용 시스템 개발을 목표를 할 경우,  
 프로그램 기간과 내용을 조정할 수 있습니다. 이  
 과정의 경우 주니어 SW 엔지니어 자격증을 획득할  
 수 있습니다.

# 6.7 요약

- ✓ 표를 그릴 때, 프리젠테이션 도구가 제공하는 기본 기능을 이용하면 세밀한 조정이 어렵습니다.
- ✓ 마찬가지, 엑셀로 부터 “개체 삽입”을 통해서 추가해도 표현 측면에서 세밀한 조정이 안됩니다.
- ✓ 이럴 경우, 사각형을 쌓아서 원하는 모양의 표를 그리는 것이 가장 효과적입니다.
- ✓ **기본 세트만 잘 만든다면, 나머지는 “복사하여 붙여넣기”이므로 생각보다 쉽게 복잡한 표를 그릴 수 있습니다.**

워크 ーシ성 지향 Java	B1. 객체모델링 UML/객체 모델링 EA/Togather/ RSA DDD	C1. 마크업 언어 HTML/XHTML CSS XML/XSLT JSON	D1.HTML 5 HTML5 JS API WebSocket Socket.io WebGL, Canvas와	E1. 패턴 문서 패턴 디자인 패턴 아키텍처 패턴 디버깅	F1.iOS Objective C Mac OS/iOS Cocoa touch 디버깅	G1.TDD TDD 단위테스트 JUnit/dbUnit	H1.SW테스트 SW 테스트 테스팅 프로세스 테스트 관리	I1. 기능테스트 기능테스트 Fit/Selenium JMeter	J1. 프로젝트관리 프로젝트관리 PMBOK QA/Audit CMMI
A2.C/C++ C/C++ C/C++ 네트워크 C/C++ 동시성 객체지향 C++	B2. 자료구조 자료구조 알고리즘 표현식 문자세트	C2. 플랫폼 - POJO Spring	D2. 웹아키텍처 웹아키텍처 Apache/jetty HTTP 이해	E2. 서버 설계 채널설계 컴포넌트 설계 플랫폼 설계	F2. 안드로이드 Android Android design 안드로이드 패턴	G2. 데이터모델링 데이터모델링 DB 최적화 SQL	H2.RDBMS 오라클 mySQL/HSQSL MS-SQL	I2. 빌드자동화 동일빌드(Maven) Nexus CI(Jenkins)	J2. 개발프로세스 UP/RUP Agile SCRUM/XP Kanban
A3.C#/.Net C# C# 네트워크 C# 동시성 객체지향 C#	B3. 요구사항 요구사항 명세서 UseCase User Story 요구사항 워크샵	C3. 플랫폼-OSGi OSGi Equinox, Felix Virgo, Aries Karaf	D3. 모바일웹 Responsive Design PhoneGap Titanium	E3. 보안 OAuth/SSO Spring Security Identity/Access 데이터 보안	F3. SW아키텍처 SW 아키텍처 아키텍팅 프로세스 아키텍처 문서화 IEEE 1471	G3. 빅데이터 빅데이터 개요 빅데이터 접근방법 빅데이터 사례연구	H3. NoSQL DB Neo4J MongoDB HBase	I3. 코드진단 코드 규약 코드 매트릭 의존성 분석 스타일 체크	J3. 조직 리더십 조직관리 조직 심리학 창의력과 혁신
A4.Javascript JavaScript jQuery Sencha Js 패턴	B4. 형상관리 SVN/SVNKit Git	C4. 플랫폼-J2EE Servlet/EJB Tomcat JBoss/Glassfish J2EE Blueprint	D4. 웹프레임워크 웹프레임워크 Struts Spring MVC	E4. 데이터프레임워크 JPA/myBATIS Hibernate iBATIS Spring Data	F4. 미리요소 설계 웹프레임워크 연계프레임워크 통신프레임워크 변환유필리티	G4. 리치클라이언트 Rich client GWT SWT/플러그인 RAP	H4. 빅데이터 분석 Hadoop 맵리듀스 R	I4. 비즈니스 제안서 작성법 제안프로세스 이해 영업프로세스 이해	J4. 논리 논리적인 사고 논리적인 주론 논리적인 토론 논리적인 글쓰기
A5. 서버스크립트 JSP/Servlet Node.js PHP, Ruby Python/ASP.Net	B5. 플랫폼 - 리눅스 Linux OS 시스템 프로그래밍	C5. 플랫폼-클라우드 클라우드 IaaS/AWS PaaS/GAE	D5. 웹기획 IA 기획 WireFrame Storyboard	E5. REST REST Jersey/Restlet REST API 설계 Open API	F5. 웹서비스 웹서비스 SOAP WSDL UDDI	G5. EAI SOA EAI/ESB/SCA MOM ActiveMQ	H5. 비즈니스 기술 워크플로우 BPM BPMN BABOK REA	I5. 비즈니스모델링 비즈니스 모델링 BPMN BABOK REA	J5. 표현 프리젠테이션 기술문서 아이

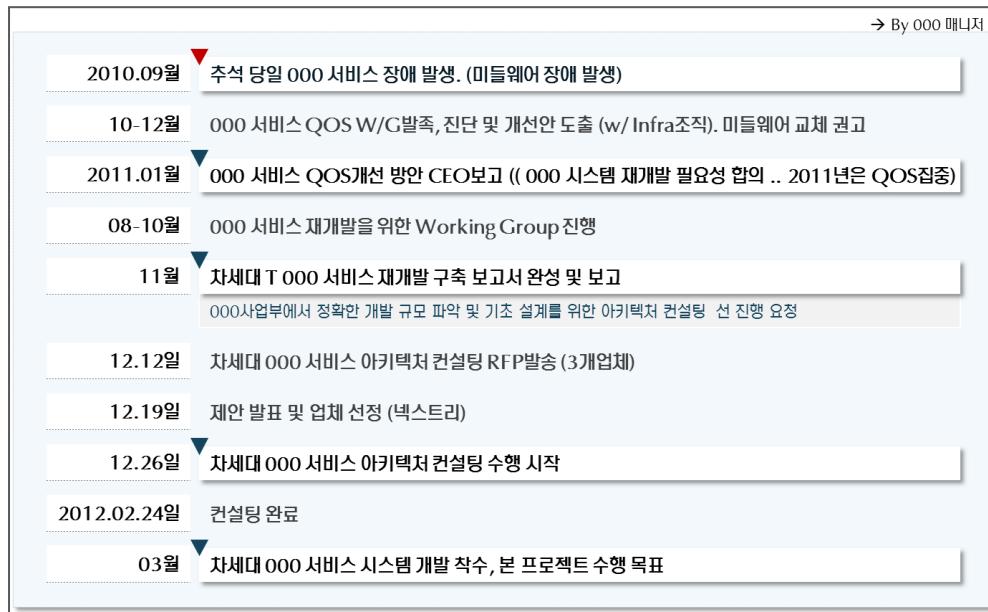


## 7. 실습 #4 – 텍스트 표현

- 
- 7.1 텍스트 표현
  - 7.2 예제: 경과 보고
  - 7.3 예제: 강의 일정
  - 7.4 요약

## 7.1 텍스트 표현

- ✓ 절차, 일정 등을 표현하는 텍스트를 텍스트 그대로 사용하는 것 보다는 약간의 그리기를 통해 전달 효과를 높일 수 있습니다.
- ✓ 그리기는 사각형 박스에 텍스트를 담는 단순한 형태입니다. 약간의 포인트를 주기 위해 작은 도형을 사용합니다.
- ✓ 하나의 정보를 어떻게 표현하느냐에 따라 전달 효과의 차이는 아주 클 수 있습니다.
- ✓ 이 장에서 실습하려는 텍스트 표현이 극명한 차이를 보여 주는 대표적인 예입니다. ← 텍스트에 도형을 입혀 봅니다.



- 1010.09월 – 추석 당일 000 서비스 장애 발생
- 10-12월 - 000 서비스 QOS W/G발족, 진단 및 개선안 도출 (w/ Infra조직). 미들웨어 교체 권고
- 2011년 1월 - 000 서비스 QOS개선 방안 CEO보고 (( 000 시스템 재개발 필요성 합의 .. 2011년은 QOS집중))
- 08-10월 - 000 서비스 재개발을 위한 Working Group 진행
- 11월 - 차세대 T 000 서비스 재개발 구축 보고서 완성 및 보고
- 12.12일 – 차세대 000 서비스 아키텍처 컨설팅 RFP 발송 (3 개 업체)
- 12.19일 – 차세대 000 서비스 아키텍처 컨설팅 수행 시작
- 2012.2.24일 – 컨설팅 완료
- 3.1일 – 차세대 000 서비스 시스템 개발 착수, 본 프로젝트 수행 목표

## 7.2 예제: 경과 보고 – 텍스트 원본

---

- ✓ 1010.09월 – 추석 당일 000 서비스 장애 발생
- ✓ 10-12월 - 000 서비스 QOS W/G발족, 진단 및 개선안 도출 (w/ Infra조직). 미들웨어 교체 권고
- ✓ 2011년 1월 - 000 서비스 QOS개선 방안 CEO보고 (( 000 시스템 재개발 필요성 합의 .. 2011년은 QOS집중)
- ✓ 08-10월 - 000 서비스 재개발을 위한 Working Group 진행
- ✓ 11월 - 차세대 T 000 서비스 재개발 구축 보고서 완성 및 보고
- ✓ 12.12일 – 차세대 000 서비스 아키텍처 컨설팅 RFP 발송 (3 개 업체)
- ✓ 12.19일 – 차세대 000 서비스 아키텍처 컨설팅 수행 시작
- ✓ 2012.2.24일 – 컨설팅 완료
- ✓ 3.1일 – 차세대 000 서비스 시스템 개발 착수, 본 프로젝트 수행 목표

## 7.2 예제: 경과 보고 – 텍스트 표현

→ By 000 매니저

2010.09월

추석 당일 000 서비스 장애 발생. (미들웨어 장애 발생)

10-12월

000 서비스 QOS W/G발족, 진단 및 개선안 도출 (w/ Infra조직). 미들웨어 교체 권고

2011.01월

000 서비스 QOS개선 방안 CEO보고 (( 000 시스템 재개발 필요성 합의 .. 2011년은 QOS집중)

08-10월

000 서비스 재개발을 위한 Working Group 진행

11월

차세대 T 000 서비스 재개발 구축 보고서 완성 및 보고

000사업부에서 정확한 개발 규모 파악 및 기초 설계를 위한 아키텍처 컨설팅 선 진행 요청

12.12일

차세대 000 서비스 아키텍처 컨설팅 RFP발송 (3개업체)

12.19일

제안 발표 및 업체 선정 (넥스트리)

12.26일

차세대 000 서비스 아키텍처 컨설팅 수행 시작

2012.02.24일

컨설팅 완료

03월

차세대 000 서비스 시스템 개발 착수, 본 프로젝트 수행 목표

## 7.3 예제: 강의 일정 – 텍스트 원본

---

### ✓ 월요일(28일)

- HTML5 소개
- Semantic Element, HTML5 API – Canvas, SVG, Audio & Video, WebStorage

### ✓ 화요일(29일)

- 웹 트랜드
- HTML5 API - WebWorker
- CSS3 소개, Responsive WebDesign, jQuery 소개

### ✓ 수요일(30일)

- UX 이해
- UI/UX 프로세스 이해 및 실습

### ✓ 목요일(31일)

- UI 기획도구 소개
- Axure RP – Pages, Widget & Master, Interaction, Dynamic Panel

### ✓ 금요일(1일)

- 프리젠테이션 개요
- 프리젠테이션 – 발표, 레이아웃, 선, 색, 놓도, 박스와 연결, 테이블

# 7.3 예제: 강의 일정 – 텍스트 표현

## 강의 순서

### 현대 UI의 핵심 – HTML5

월 (28)

- HTML5 소개
- Semantic Element
- HTML5 API
- HTML5 API – Canvas, SVG
- HTML5 API – Audio & Video
- HTML5 API - WebStorage

### 현대 UI의 핵심 – CSS3, Javascript

화 (29)

- HTML5 API – WebWorker
- CSS3 소개
- Responsive WebDesign
- jQuery 소개
- **Web Front-End 트렌드**
- 질의 응답

### UI를 넘어서 UX 이해

수 (30)

- UX 이해
- UI/UX 프로세스
- 실습: UX 프로세스 따라하기

### UI 기획 실무 1 – UI 프로토타입 도구

목 (31)

- UI 기획 도구(Axure RP) 소개
- Axure RP - Pages
- Axure RP - Widget & Master
- Axure RP – Interaction
- Axure RP – Dynamic Panel
- Axure RP – Condition, Values

### UI 기획 실무 II – 파워포인트를 이용한 프리젠테이션

금 (1)

- 프리젠테이션 개요
- 프리젠테이션 – 발표
- 프리젠테이션 자료 – 레이아웃
- 프리젠테이션 자료 – 선, 색, 놓도
- 프리젠테이션 자료 – 박스와 연결
- 프리젠테이션 자료 – 테이블

## 7.4 요약

- ✓ 일정, 경과 등을 보여 주는 텍스트는 텍스트의 의미에 시간의 흐름, 중요성 등이 포함되어 있습니다.
- ✓ 이러한 텍스트는 한 줄 한 줄 의미를 담고 있습니다. 텍스트에 도형을 입혀야 하는 이유입니다.
- ✓ 텍스트에 적절하게 도형을 입히고 나면, 내용 전달 효과가 훨씬 더 높아집니다.
- ✓ 이런 작업은 하나의 자료에 한 두 페이지로 제한합니다. 모든 텍스트를 이런 식으로 표현한다면 청중의 피로감을 높입니다.

→ By 000 매니저	
2010.09월	추석 당일 000 서비스 장애 발생. (미들웨어 장애 발생)
10-12월	000 서비스 QOS W/G발족, 진단 및 개선안 도출 (w/ Infra조직). 미들웨어 교체 권고
2011.01월	000 서비스 QOS개선 방안 CEO보고 (( 000 시스템 재개발 필요성 합의 .. 2011년은 QOS집중)
08-10월	000 서비스 재개발을 위한 Working Group 진행
11월	차세대 T 000 서비스 재개발 구축 보고서 완성 및 보고 000사업부에서 정확한 개발 규모 파악 및 기초 설계를 위한 아키텍처 컨설팅 선 진행 요청
12.12일	차세대 000 서비스 아키텍처 컨설팅 RFP발송 (3개업체)
12.19일	제안 발표 및 업체 선정 (넥스트리)
12.26일	차세대 000 서비스 아키텍처 컨설팅 수행 시작
2012.02.24일	컨설팅 완료
03월	차세대 000 서비스 시스템 개발 착수, 본 프로젝트 수행 목표

- 1010.09월 – 추석 당일 000 서비스 장애 발생
- 10-12월 - 000 서비스 QOS W/G발족, 진단 및 개선안 도출 (w/ Infra조직). 미들웨어 교체 권고
- 2011년 1월 - 000 서비스 QOS개선 방안 CEO보고 (( 000 시스템 재개발 필요성 합의 .. 2011년은 QOS집중)
- 08-10월 - 000 서비스 재개발을 위한 Working Group 진행
- 11월 - 차세대 T 000 서비스 재개발 구축 보고서 완성 및 보고
- 12.12일 – 차세대 000 서비스 아키텍처 컨설팅 RFP 발송 (3 개 업체)
- 12.19일 – 차세대 000 서비스 아키텍처 컨설팅 수행 시작
- 2012.2.24일 – 컨설팅 완료
- 3.1일 – 차세대 000 서비스 시스템 개발 착수, 본 프로젝트 수행 목표

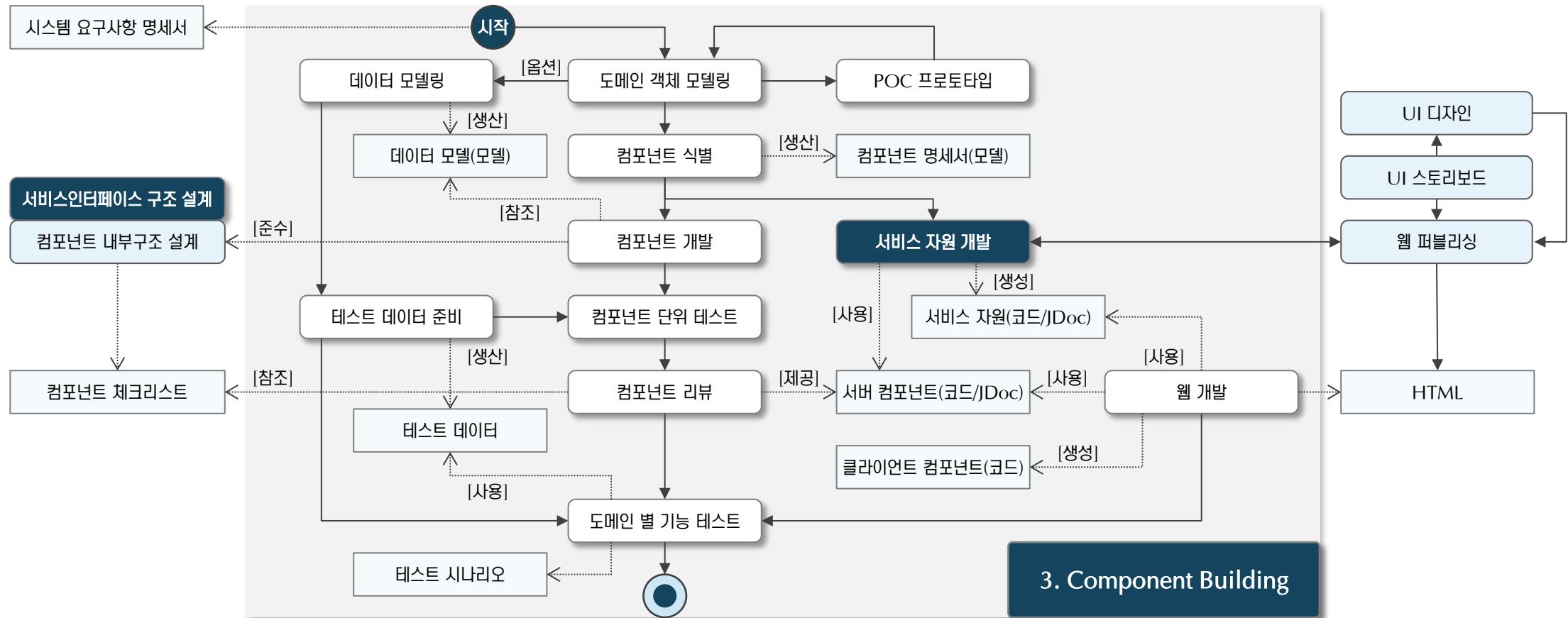


## 8. 실습 #5 – 다이어그램과 그래프

- 
- 8.1 복잡한 다이어그램 표현
  - 8.2 복잡한 그래프 표현
  - 8.3 세밀한 절차 표현
  - 8.4 정성을 들인 다이어그램 표현
  - 8.5 간결한 다이어그램 표현
  - 8.6 요약

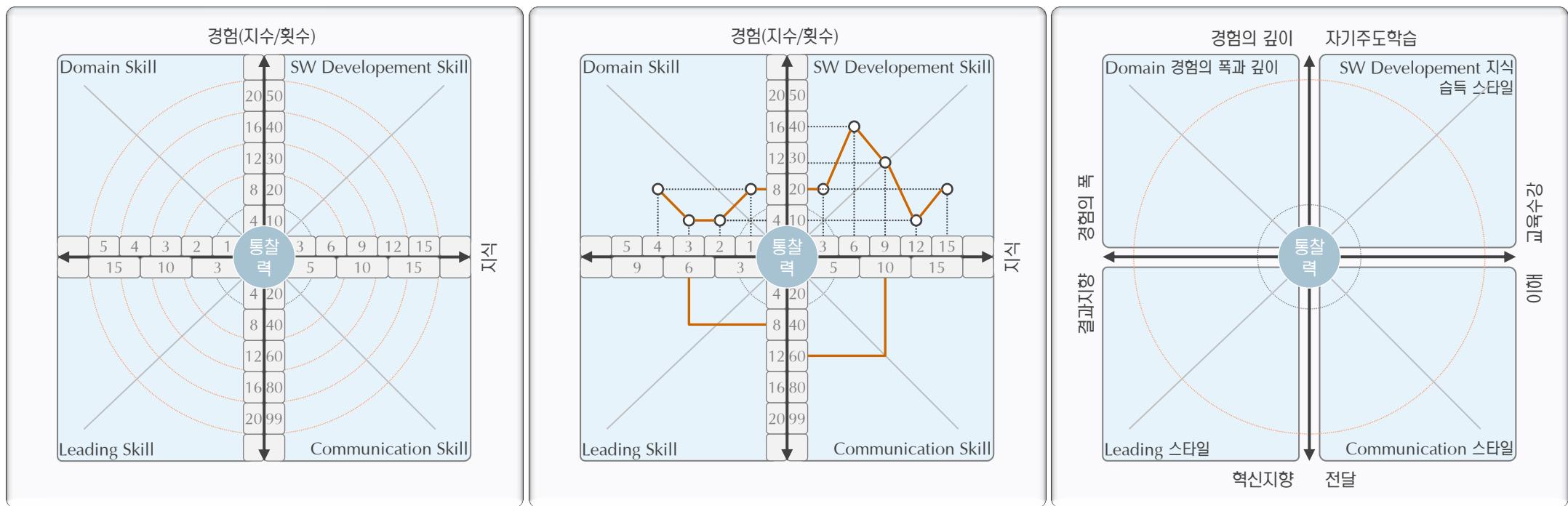
## 8.1 복잡한 다이어그램 표현

- ✓ 표현(representation)에 이만큼의 노력을 들일 이유가 있을까요?
  - ✓ 표현하는 작업은 다른 유형의 설계 활동이며, 그런 관점에서 표현 도구는 모델링 도구 역할을 합니다.
  - ✓ 표현을 해야 전달할 수 있고, 전달을 해야 피드백을 받을 수 있으며, 피드백을 받을 수 있어야 좋은 설계가 가능합니다.



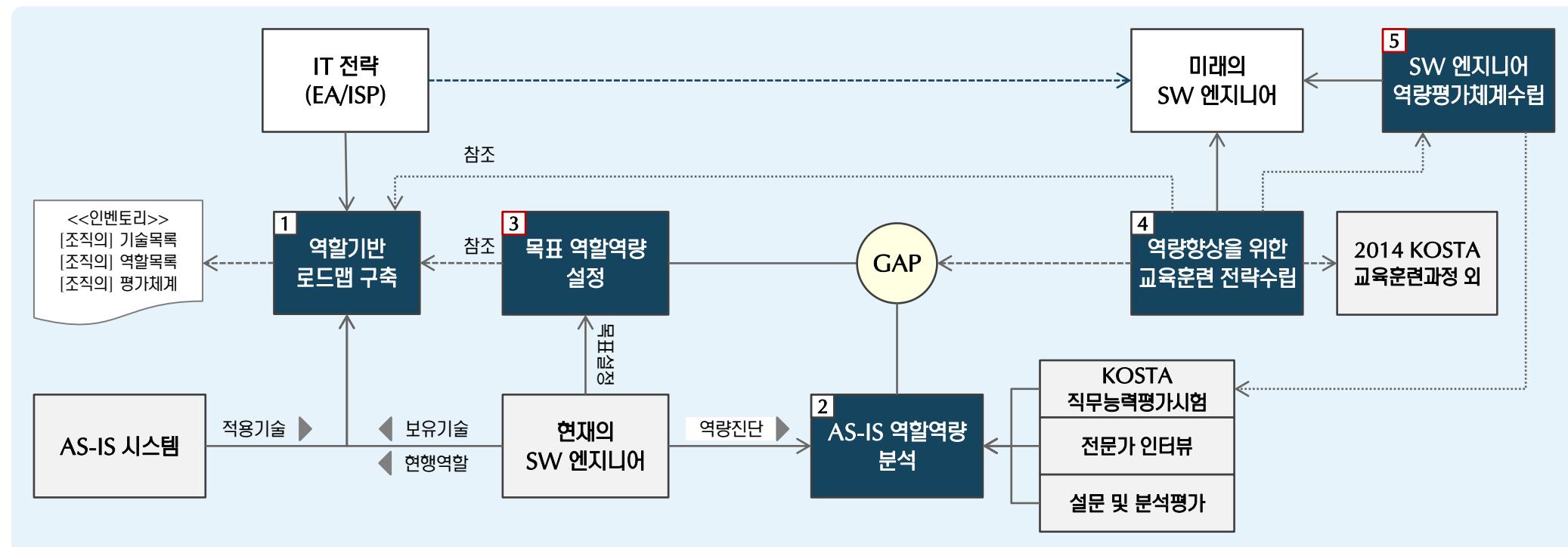
## 8.2 복잡한 그래프 표현

- ✓ 눈금 및 안내선 창에서 눈금 설정 – 간격은 팀에서 통일해야 합니다. 0.2cm를 추천합니다.
- ✓ 객체 하나만 제대로 그리면, 나머지는 복사하여 붙이기 수준이므로 처음 하나를 그릴 때 정성을 다합니다.
- ✓ 세밀한 조정( Control + 화살표 키)은 가급적 하지 않습니다. 복사하여 다른 곳으로 이동할 때 그림이 깨어집니다.
- ✓ 사각형은 정확히 맞출 수 있지만 원은 맞추기 힘듭니다. 세밀한 조정을 하지 않는 선에서 도형을 마무리 짓습니다.



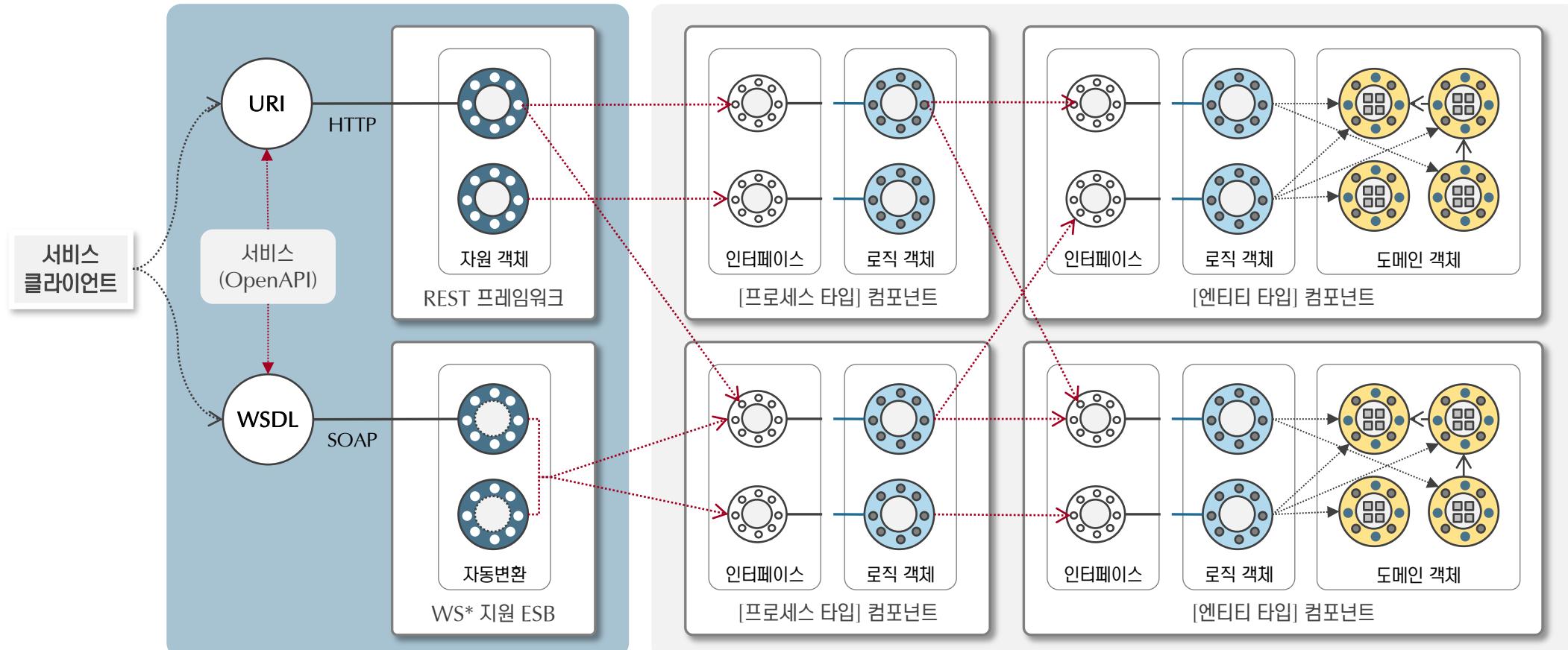
### 8.3 세밀한 절차 표현

- ① 기존에 시스템 구축에 사용한 기술, 엔지니어 현재 기술, 조직의 미래 IT 전략을 참조하여 조직에 최적화된 로드맵을 구축합니다.
  - ② SW 엔지니어의 현재 역량 특성을 분석하여, 수행 가능한 역할과 해당 역할 적합도(goodness of fit)를 산정합니다.
  - ③ 현재 수행 가능한 역할(acceptable role)을 기준으로 올해의 목표 역할을 설정하여, 개인/팀/회사의 목표를 설정합니다. ← 방향설정
  - ④ 현재 역할(present role)에서 목표 역할(planned role) 사이의 차이를 메울 수 있도록 교육훈련방안을 제시합니다.
  - ⑤ 객관적으로 타당한 역량 평가체계를 수립하고, 평가체계를 통해서 엔지니어가 성장해 나갈 방향을 제시합니다. ← 방향설정



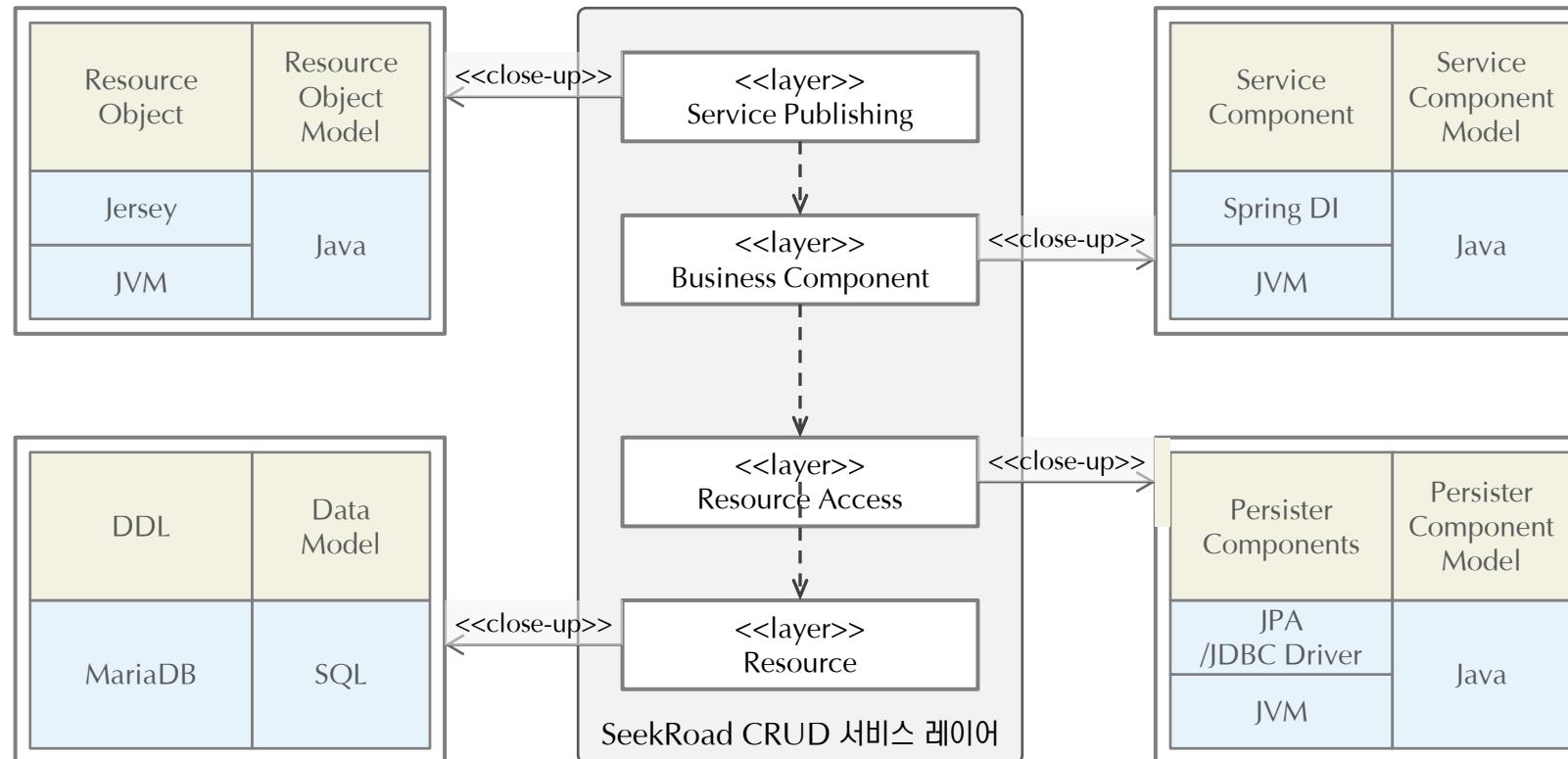
## 8.4 정성을 들인 다이어그램 표현

- ✓ 객체, 컴포넌트, 서비스 이 세 가지 설계 요소가 서로 어떤 관계가 있는지 설명할 필요가 있었습니다.
- ✓ 적당한 표현은 서로 다른 생각을 가진 사람들 간에 혼란을 부추길 수 있는 상황입니다.
- ✓ 기술적인 내용이지만 기술을 모르는 이해관계자(stakeholder)도 설득할 수 있도록 자료를 준비했습니다.
- ✓ 다음 그림은 논쟁의 핵심인 “SOA에서 서비스의 정체”를 밝히는데 충분한 자료가 되었습니다.



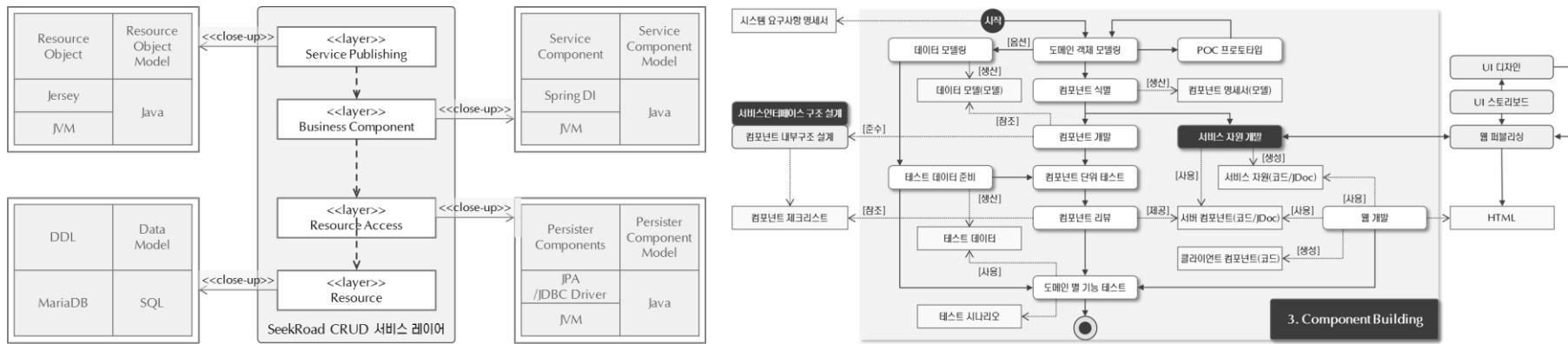
## 8.5 간결한 다이어그램 표현

- ✓ 아래 다이어그램은 어떤 서비스 서버(sub-system)의 논리적인 구조를 layered style로 보여 주고 있습니다.
- ✓ 색상에 자신이 없을 경우, 흰색을 그대로 두거나 주변과 차별화를 해야 할 경우, 옅은 회색으로 처리합니다.
- ✓ 이 다이어그램 역시 기본 세트를 완성한 후, “복사하여 붙이기” 과정을 통해 완성합니다.
- ✓ 사각형 박스 안에 들어 가는 텍스트 폰트 크기를 세밀하게 조절하여 줍니다.



## 8.6 요약

- ✓ 복잡한 다이어그램의 첫 걸음은 하나의 도형을 그리고 조정하는 작업으로부터 시작합니다.
  - ✓ 하나의 기준 도형이 완성되면, 그 도형을 이용하여 기준 세트를 만듭니다.
  - ✓ 기준 세트가 완성되면, 머리에 구상했던 그림을 하나 둘 현실로 만들어 갑니다.
  - ✓ 모든 단계에서 항상 크기, 위치에 대한 조정 작업을 지속적으로 하여야, 페이지에 어울리는 멋진 결과를 얻을 수 있습니다.





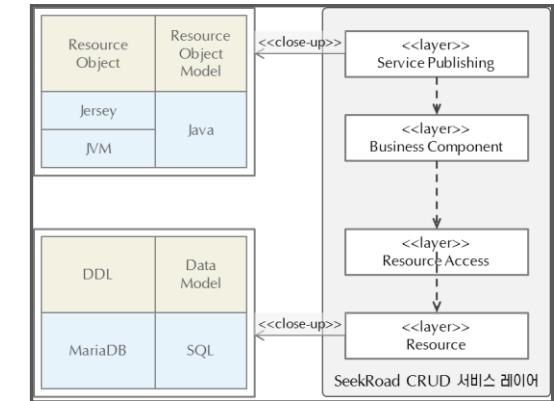
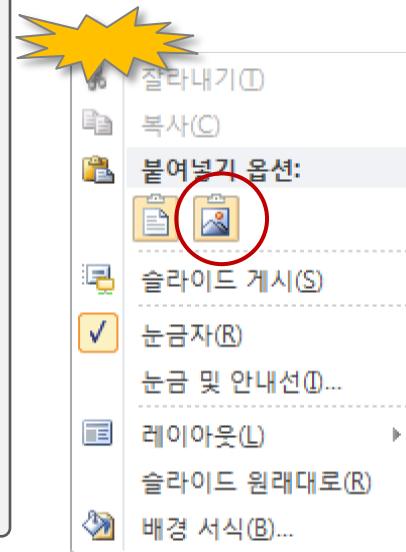
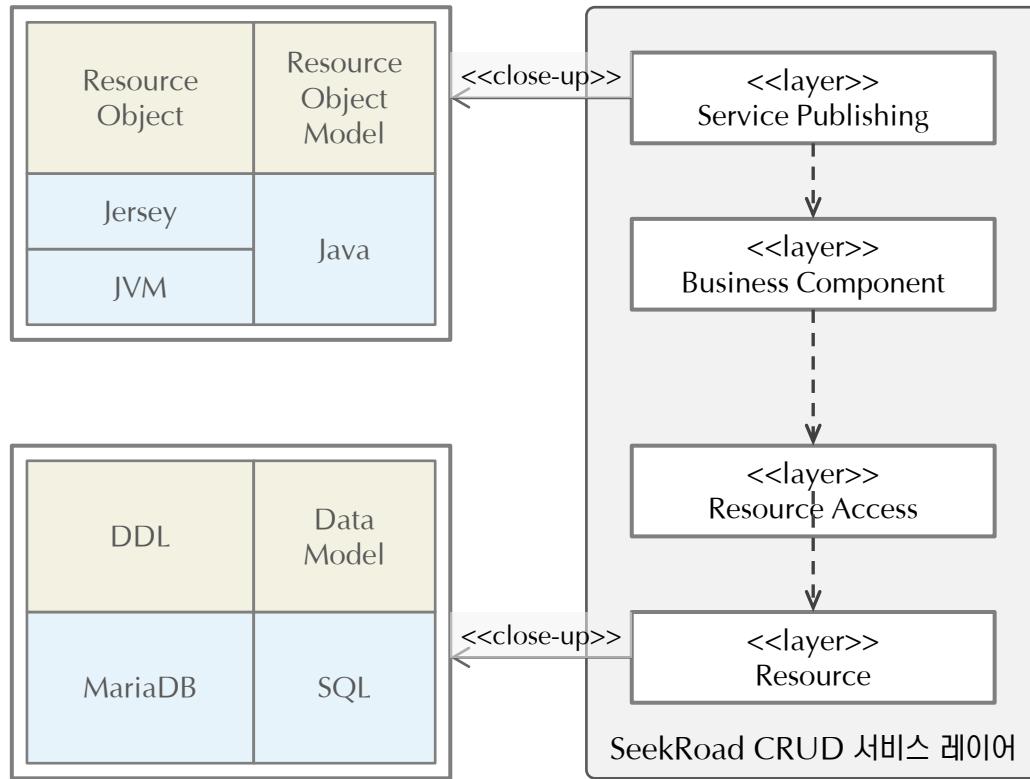
## 9. 유용한 팀

- 9.1 복사하여 이미지로 붙이기
- 9.2 개요와 목차 구성
- 9.3 템플릿 – 페이지 크기 조정
- 9.4 페이지 복사 후 레이아웃 지정
- 9.5 복사하여 텍스트만 붙이기
- 9.6 눈금 설정 변경 및 미세 조정
- 9.7 템플릿 기반 서식 조정
- 9.8 페이지 번호 추가하기
- 9.9 이미지 자르기

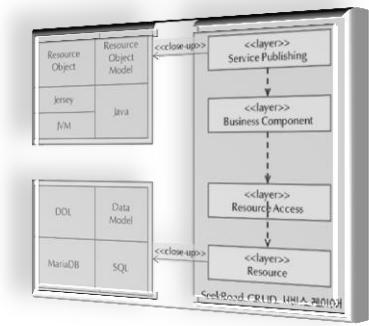
- 9.10 고화질 이미지
- 9.11 서식 복사
- 9.12 텍스트 상자 음영
- 9.13 도형을 텍스트 크기에 맞춤
- 9.14 요약

# 9.1 복사하여 이미지로 붙이기

- ✓ 도형으로 그린 그림을 이미지로 저장하면 그림에 다양한 효과를 주는 등 쉽게 다룰 수 있습니다.
- ✓ 이럴 경우 “복사한 후, 그림으로 붙여 넣기”를 하면 매우 편리합니다.
- ✓ 컨텍스트 메뉴(오른쪽 버튼 클릭) → “붙여 넣기 옵션 → 그림”을 선택합니다.

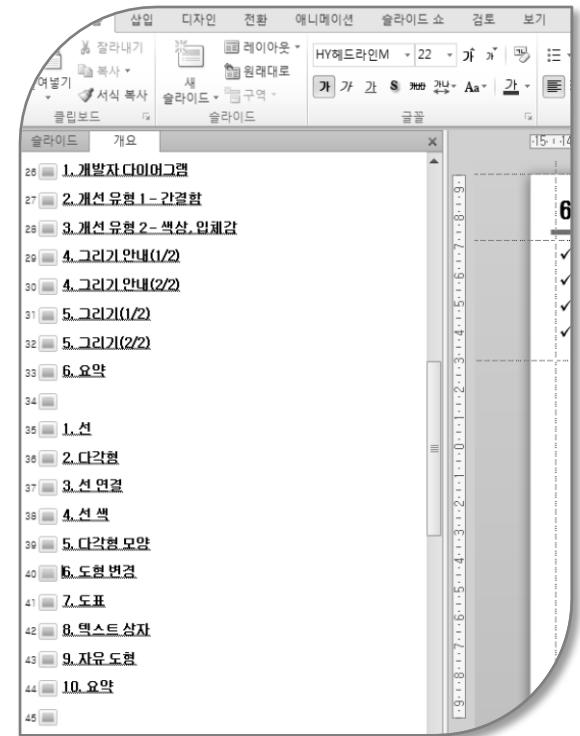
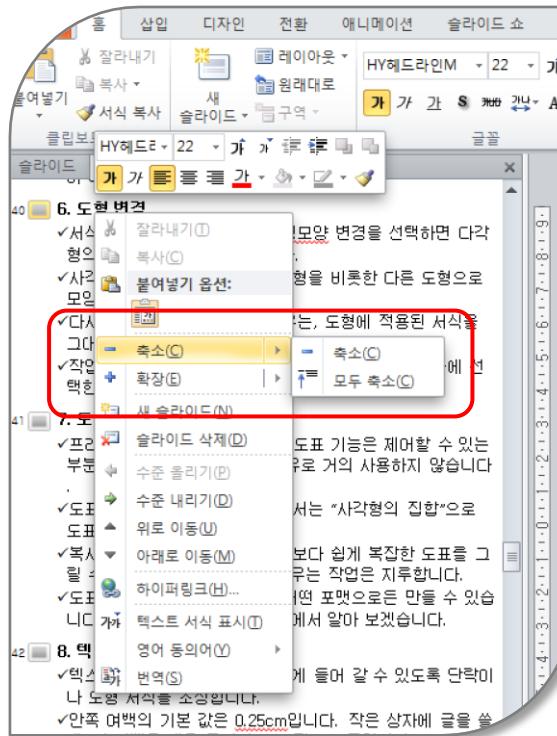
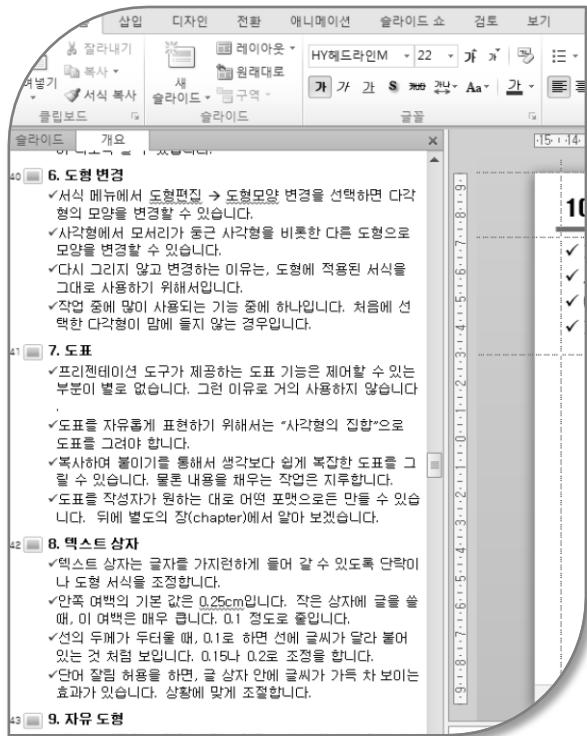


이미지로 복사



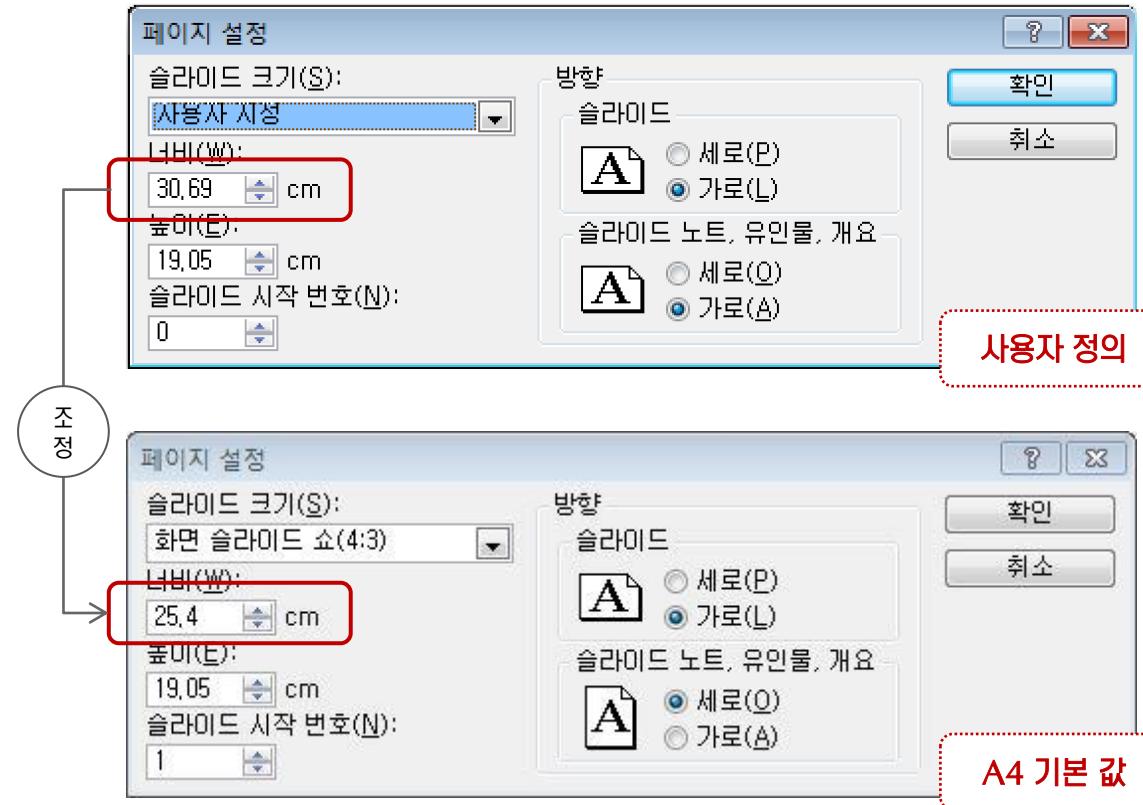
## 9.2 개요와 목차 구성

- ✓ 페이지 단위로 넘기면서, 페이지 타이틀을 조정하는 일은 매우 지루하며, 실수를 하기 쉬운 상황입니다.
- ✓ 이때 개요 기능을 활용합니다. 좌측 슬라이드 옆, 개요 선택, 축소 → 모두 축소를 선택합니다.
- ✓ 그러면 페이지의 제목을 한 눈에 보면서 시나리오를 조정할 수 있습니다.
- ✓ 목차를 편집할 때도, 개요(모두 축소 모드)를 보면서 편집할 수 있습니다.



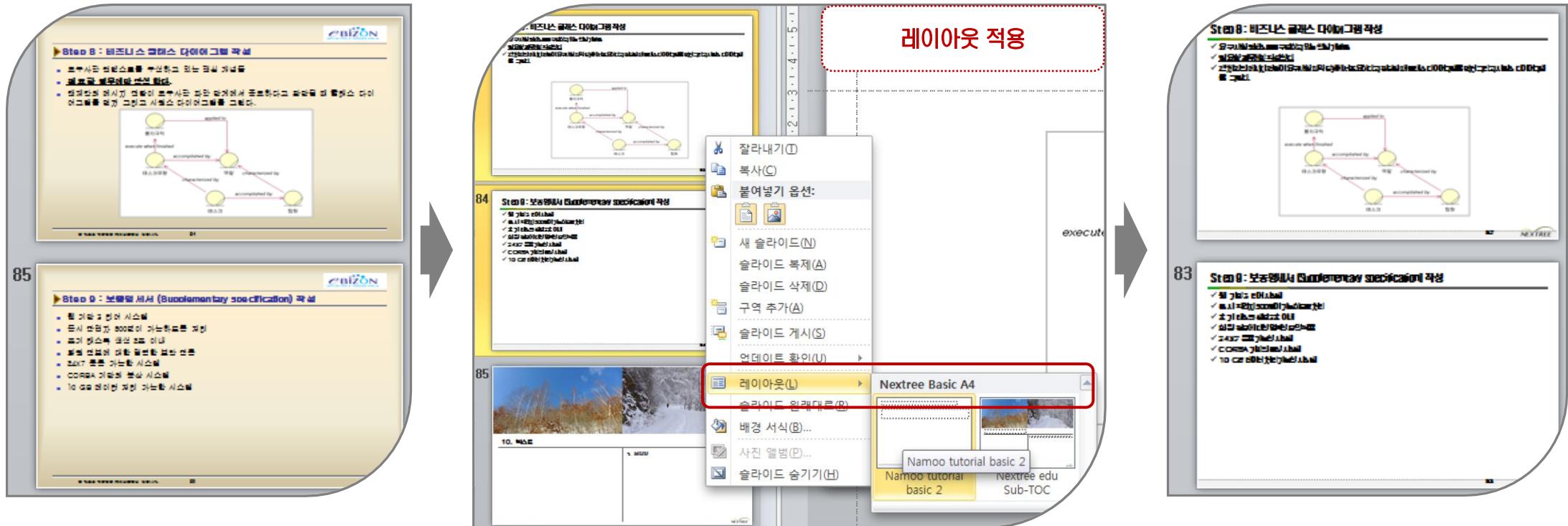
## 9.3 템플릿 – 페이지 크기 조정

- ✓ 페이지 크기를 지정할 수 있습니다. 디자인 → 페이지 설정 → 너비와 높이 조정
- ✓ 본 교재는 너비를 A4 사이즈 기준인 25.4cm로 부터 확장하여 30.69cm로 조정하였습니다.
- ✓ 페이지 크기를 조정할 경우, 넓은 공간에 많은 정보를 담을 수 있지만, 프린트할 때 주의하여야 합니다.



## 9.4 페이지 복사 후 레이아웃 지정(1/3)

- ✓ 한 장 이상의 페이지를 다른 PPT 파일로부터 복사하여 와서 붙이기를 한 후,
- ✓ 컨텍스트 메뉴(오른 쪽 버튼 클릭) → 레이아웃 → 레이아웃을 선택하면, 일괄 적용 됩니다.
- ✓ 복사 소스가 되는 파일은 반드시 기본 템플릿에 맞추어서 작성한 것이어야 합니다.



## 9.4 페이지 복사 후 레이아웃 지정(2/3)

- ✓ 복사할 때, 원본 서식을 그대로 적용하면, 서식 템플릿이 함께 팔려 옵니다.
- ✓ 이렇게 되면 서식 템플릿이 넘쳐나서 서식 관리가 어려워집니다. ← 서식 템플릿은 단순할 수록 좋습니다.
- ✓ 페이지 복사 후, 곧 바로 서식을 지정하고, 서식 마스터로 들어가 팔려 온 레이아웃 서식을 확인하고 정리합니다.
- ✓ 레이아웃의 컨텍스트 메뉴에서 “레이아웃 삭제”가 활성화 되는 경우는 사용되지 않는 것이므로 삭제해도 됩니다.

**▶ Step 8 : 비즈니스 클래스 다이어그램 작성**

ebIZON  
Let's On Your e-business with us

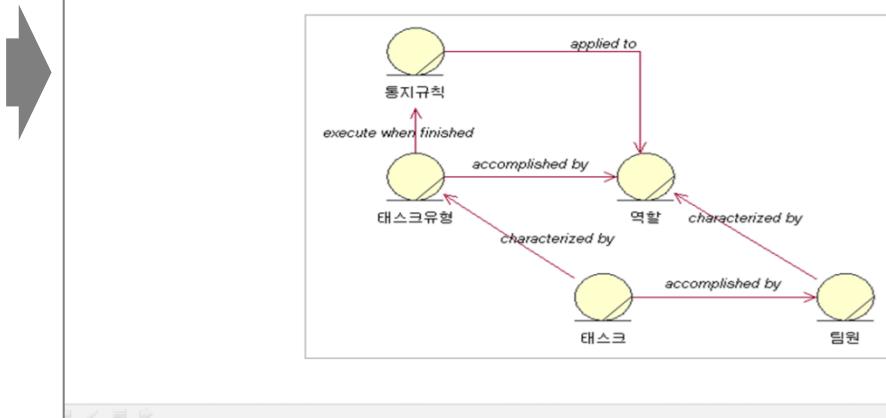
- 요구사항 컨텍스트를 구성하고 있는 핵심 개념들
- 필요할 경우에만 작성 한다.
- 객체간의 메시지 전달이 요구사항 파악 단계에서 중요하다고 판단될 때 클래스 다이어그램을 먼저 그리고 시퀀스 다이어그램을 그린다.

```
graph TD; BR((Business Rule)) -- "apply to" --> TT((Task Type)); TT -- "execute when finished" --> TT; TT -- "accomplished by" --> T((Task)); TT -- "characterized by" --> T; TT -- "characterized by" --> Team((Team)); T -- "accomplished by" --> Team;
```

본 자료의 저작권은 (주)아이비즈온에 있습니다.

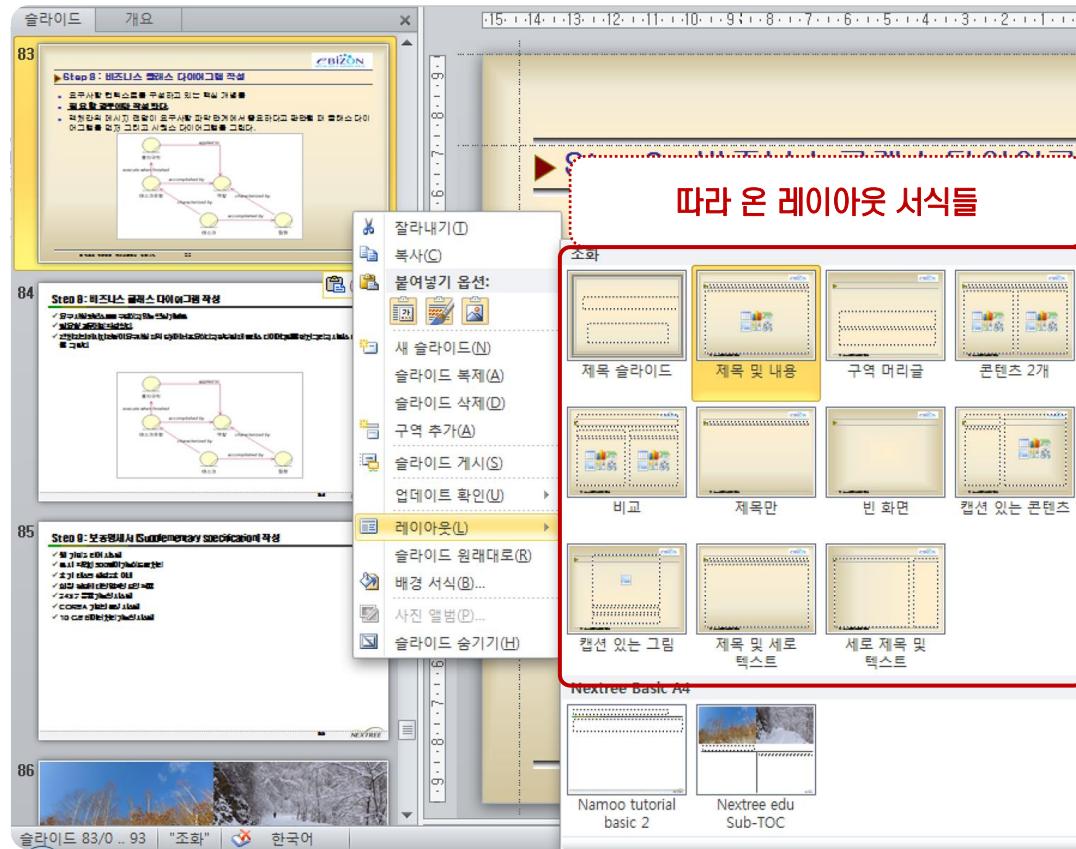
### Step 8 : 비즈니스 클래스 다이어그램 작성

- ✓ 요구사항 컨텍스트를 구성하고 있는 핵심 개념들
- ✓ 필요할 경우에만 작성 한다.
- ✓ 객체간의 메시지 전달이 요구사항 파악 단계에서 중요하다고 판단될 때 클래스 다이어그램을 먼저 그리고 시퀀스 다이어그램을 그린다.



## 9.4 페이지 복사 후 레이아웃 지정(3/3)

- ✓ 아래 그림은 다른 파일로부터 가져온 페이지를 복사한 후에 내 마스터 템플릿 내에 들어 온 레이아웃 서식을 보여 줍니다.
- ✓ 레이아웃을 확인한 후 사용하지 않는 것들은 지워 줍니다.
- ✓ 마스터 서식을 수시로 확인하고 관리하지 않으면, 이내 각종 서식들로 가득차게 됩니다.

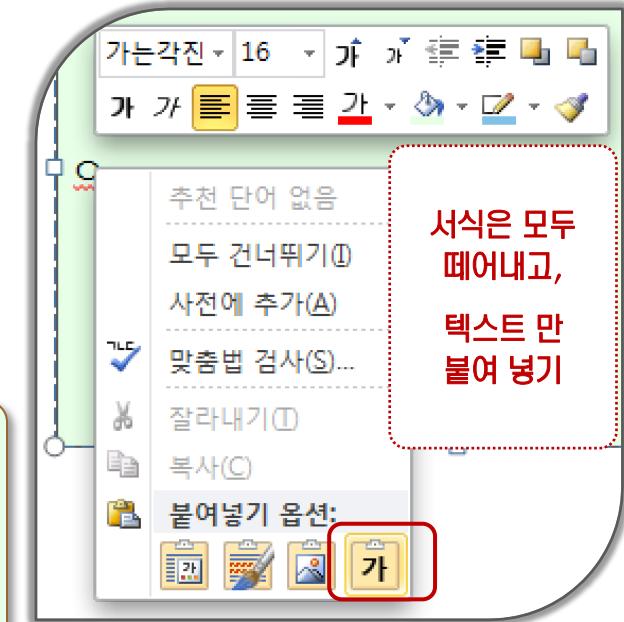


## 9.5 복사하여 텍스트만 붙이기

- ✓ 목적지 텍스트 영역에 이미 서식이 설정되어 있으며, 그 서식을 준수해야 하는 상황입니다.
- ✓ 이 경우, 복사하여 텍스트만 붙이기를 하면 서식이 없이 텍스트만 복사되어 목적지의 서식이 적용됩니다.
- ✓ 컨텍스트 메뉴에서 붙여넣기 옵션: → 마지막 메뉴를 선택합니다.

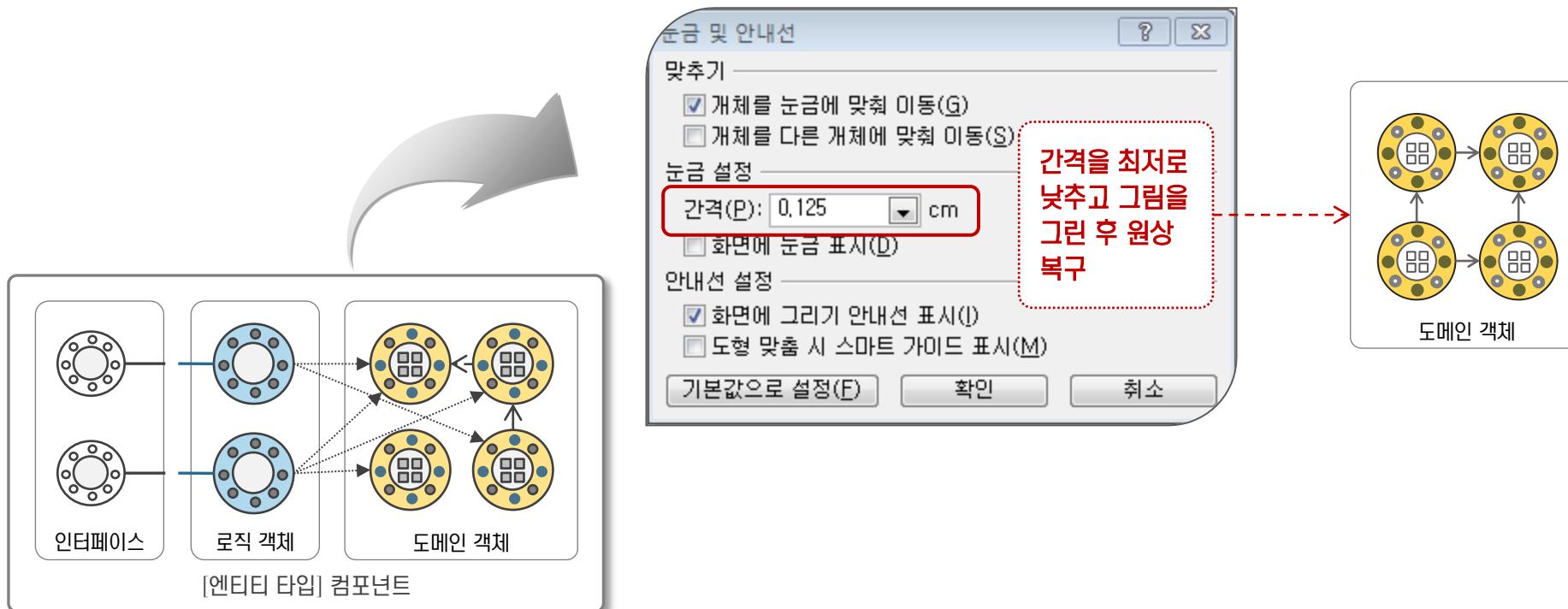
3DPoint 는 2DPoint타입이 아니기때문에 equals()를 하게 되면 false를 리턴합니다. 정확한 타입의 객체를 비교 할 수 있도록 구현해야합니다. 또 3DPoint가 2DPoint로 어떻게 표현되는지 알아야 한다면 PointasPoint()와 같은 뷰 메소드를 구현하면 됩니다. 이 모든 구현 계약을 지키는 equals()를 만드는 비법은 다음과 같습니다. (이 비법은 Joshua Bloch의 [Effective Java Programming LanguageGuide]에 나오는 것을 인용했습니다)

3DPoint 는 2DPoint타입이 아니기때문에 equals()를 하게 되면 false를 리턴합니다. 정확한 타입의 객체를 비교 할 수 있도록 구현해야합니다. 또 3DPoint가 2DPoint로 어떻게 표현되는지 알아야 한다면 PointasPoint()와 같은 뷰 메소드를 구현하면 됩니다. 이 모든 구현 계약을 지키는 equals()를 만드는 비법은 다음과 같습니다. (이 비법은 Joshua Bloch의 [Effective Java Programming LanguageGuide]에 나오는 것을 인용했습니다)



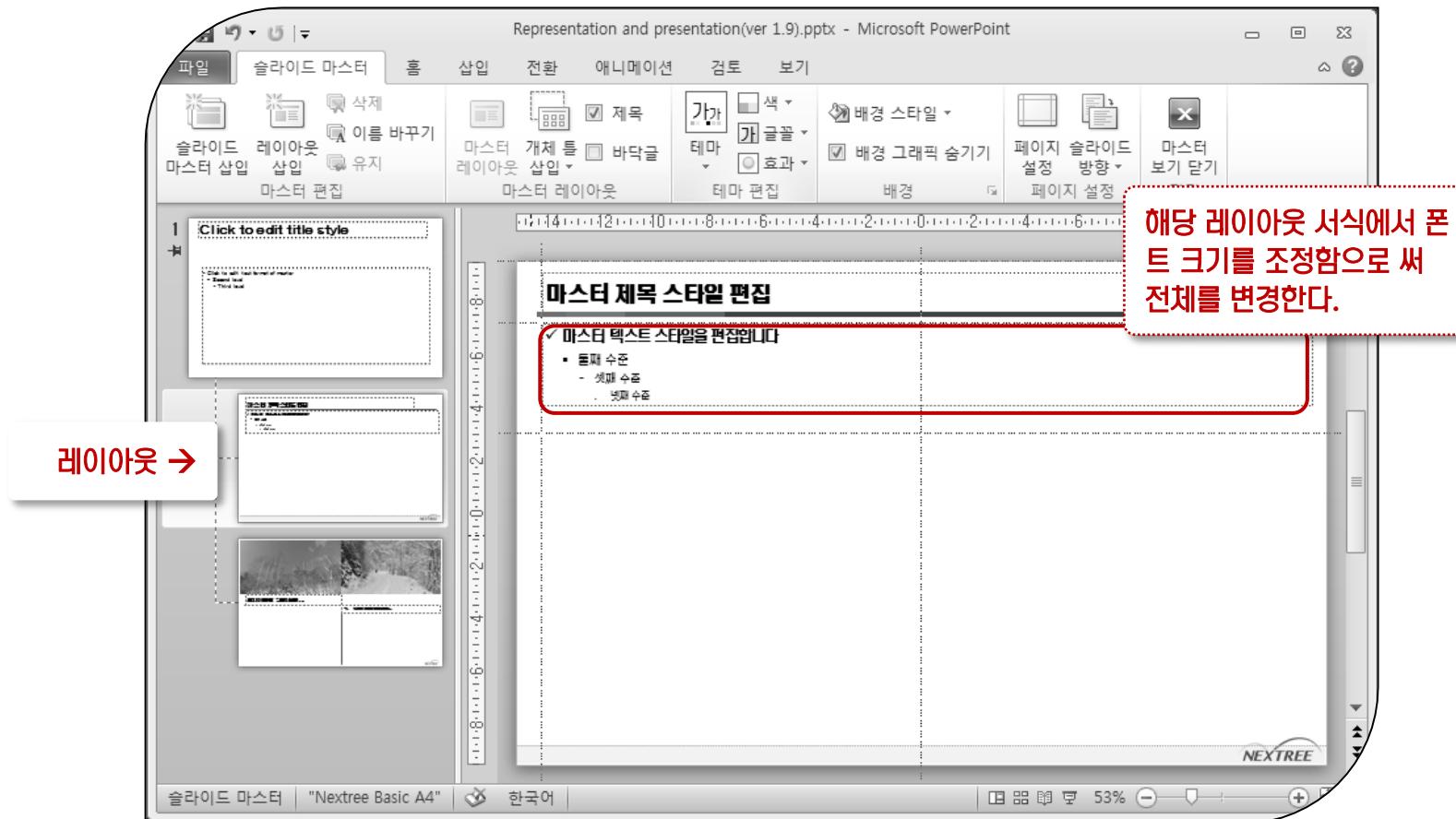
## 9.6 눈금설정 변경 및 미세 조정

- ✓ 아주 세밀한 그림을 그릴 때는 임시로 눈금 설정 → 간격을 최저 값(0.125 cm)으로 설정합니다.
- ✓ 세밀한 그림은 이 값으로도 부족하므로, control 키와 방향키를 이용하여 그림을 그립니다.
- ✓ 다 그리고 나면, 원래 눈금 설정으로 돌려 놓습니다. 이 그림을 변경할 때는 다시 간격을 최저로 조절합니다.
- ✓ 이렇게 그린 그림은 그룹으로 묶어서 이동을 합니다.



## 9.7 템플릿 기반 서식 조정

- ✓ 템플릿 내의 해당 레이아웃의 서식을 조정함으로써 전체 서식을 변경할 수 있습니다.
- ✓ 보기 → 슬라이드 마스터를 선택한 후, 해당 레이아웃 서식에서 서식 조정을 합니다.
- ✓ 전체 조정이 가능 하려면, 반드시 서식을 준수하여야 합니다.
- ✓ 서식의 텍스트 영역을 다른 텍스트 영역으로 대체하면 해당 영역은 전체 변경에서 제외됩니다.



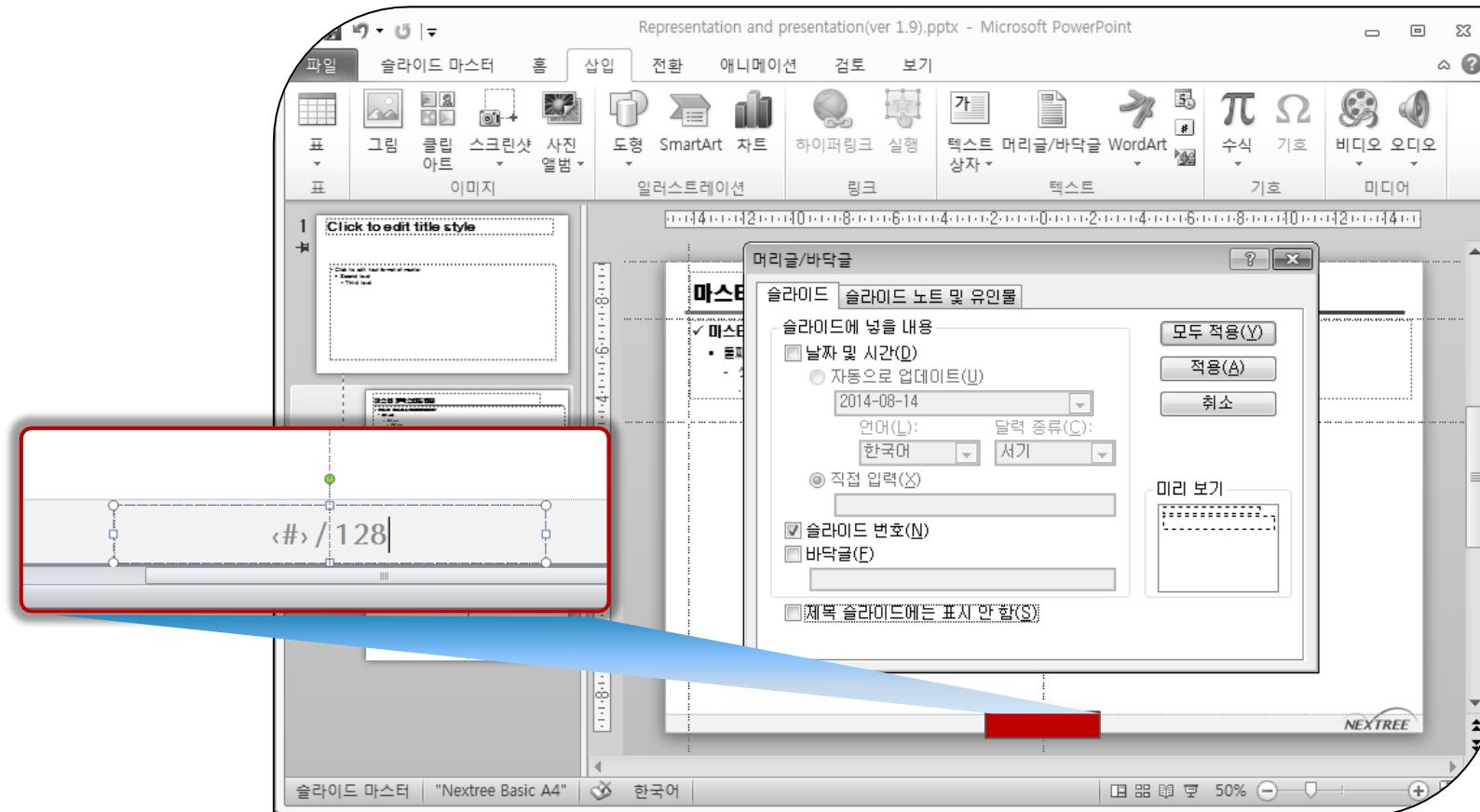
## 9.8 페이지 번호 추가하기 (1/3) – 슬라이드 번호 사용하기

---

- ✓ 템플릿에서 슬라이드 번호를 추가하면, 슬라이드의 원하는 위치에 현재 페이지 번호를 보여줄 수 있습니다.
- ✓ 템플릿에서 슬라이드 번호를 추가하는 첫 번째 방법은 [마스터 레이아웃]을 이용하는 것입니다.
- ✓ 템플릿의 슬라이드 마스터 → 마스터 레이아웃 → [슬라이드 번호]를 선택하고, 원하는 위치와 서식을 적용합니다.
- ✓ 페이지 번호를 적용할 슬라이드에서 [삽입] 탭의 [슬라이드 번호] 선택하면 페이지 번호가 슬라이드에 보여집니다.

## 9.8 페이지 번호 추가하기 (2/3) – 전체 페이지 설정

- ✓ 슬라이드에 페이지 번호를 추가하는 방법은 간단합니다.
- ✓ 보기 → 슬라이드 마스터에서 해당 레이아웃 서식을 선택하고, 삽입 → 슬라이드 번호 선택한 후 필요한 값을 추가합니다.
- ✓ 하지만 “현재 페이지 / 전체 페이지 수”, 즉 “23/128”과 같은 포맷으로 표현하고 싶은 경우가 있습니다.
- ✓ 이때는 슬라이드 마스터의 레이아웃 아래 쪽에, 아래 그림과 같이 “/ 120”과 같이 전체 페이지 수를 기록합니다.



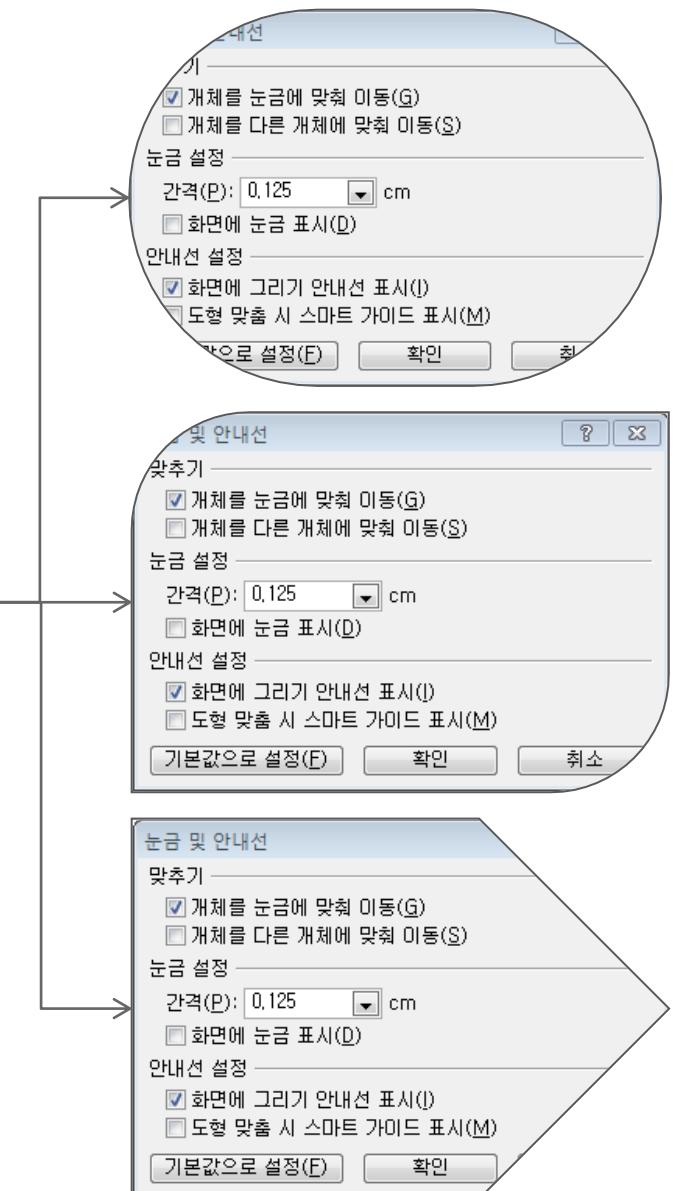
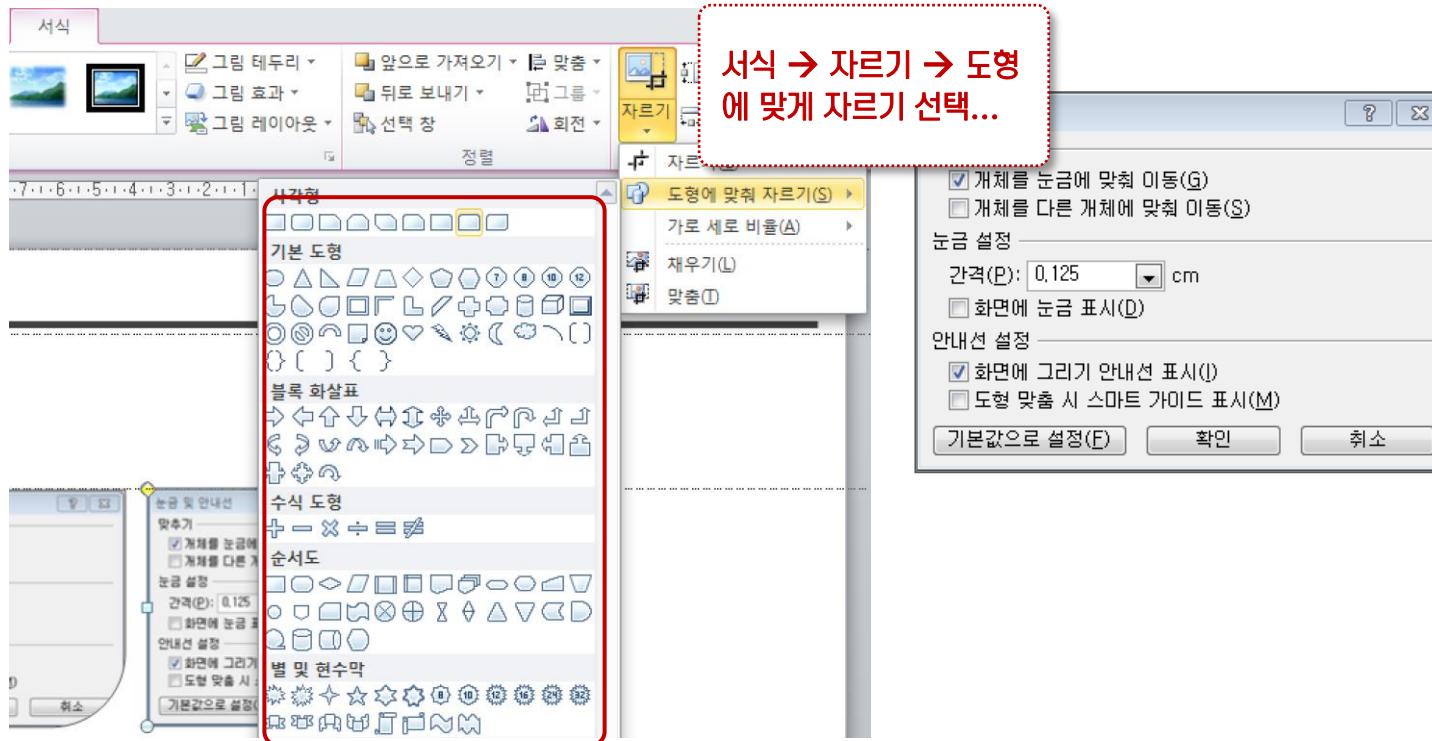
## 9.8 페이지 번호 추가하기 (3/3) – 텍스트 상자 사용하기

---

- ✓ 그러나, 마스터 레이아웃에서 슬라이드 번호를 삽입하는 방법은 슬라이드 편집 시 불편함이 있습니다.
- ✓ 슬라이드 번호 개체까지 마우스 드래그로 함께 선택되어, 의도하지 않게 슬라이드 번호 위치가 바뀌거나 삭제될 수 있습니다.
- ✓ 페이지 번호를 삽입하는 다른 방법은, 슬라이드 마스터에 텍스트 상자를 추가하여 슬라이드 번호를 삽입하는 방법입니다.
- ✓ [삽입] – [텍스트 상자]를 선택하여 원하는 위치를 정한 후, 텍스트 편집 상태에서 [삽입] → [슬라이드 번호]를 선택합니다.

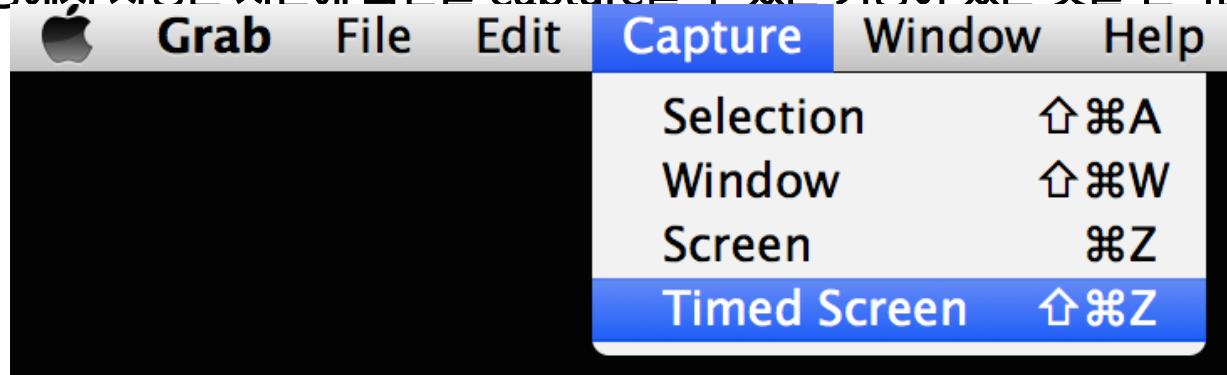
## 9.9 이미지 자르기

- ✓ 때로는 이미지를 도형에 맞추어 잘라야 할 때가 있습니다.
- ✓ 이 때 외부 이미지 도구를 사용하면 매우 번거롭습니다.
- ✓ 서식 → 자르기 → 도형에 맞게 자르기 를 선택하면 원하는 모양을 얻을 수 있습니다.
- ✓ MS Office 2007 버전은 메뉴의 위치가 다릅니다.



## 9.10 고화질 이미지(1/2)

- ✓ 컴퓨터의 화면을 capture하여 자료를 만드는데 사용하는 경우가 많습니다.
- ✓ 이 때의 문제는 낮은 해상도와 (마우스 커서가 눌려 있을 때나 대상 위에 있을 때만 동작하는) 컨텍스트 메뉴 capture입니다
- ✓
- ✓ Capture 도구를 선택할 때, 압축률은 낮지만 고화질을 지원하는 tiff 포맷을 지원하는 도구를 선택합니다.
- ✓ Timed capture가 가능해서 지정된 시간에 화면을 capture할 수 있는 기능이 있는 것을 선택합니다.

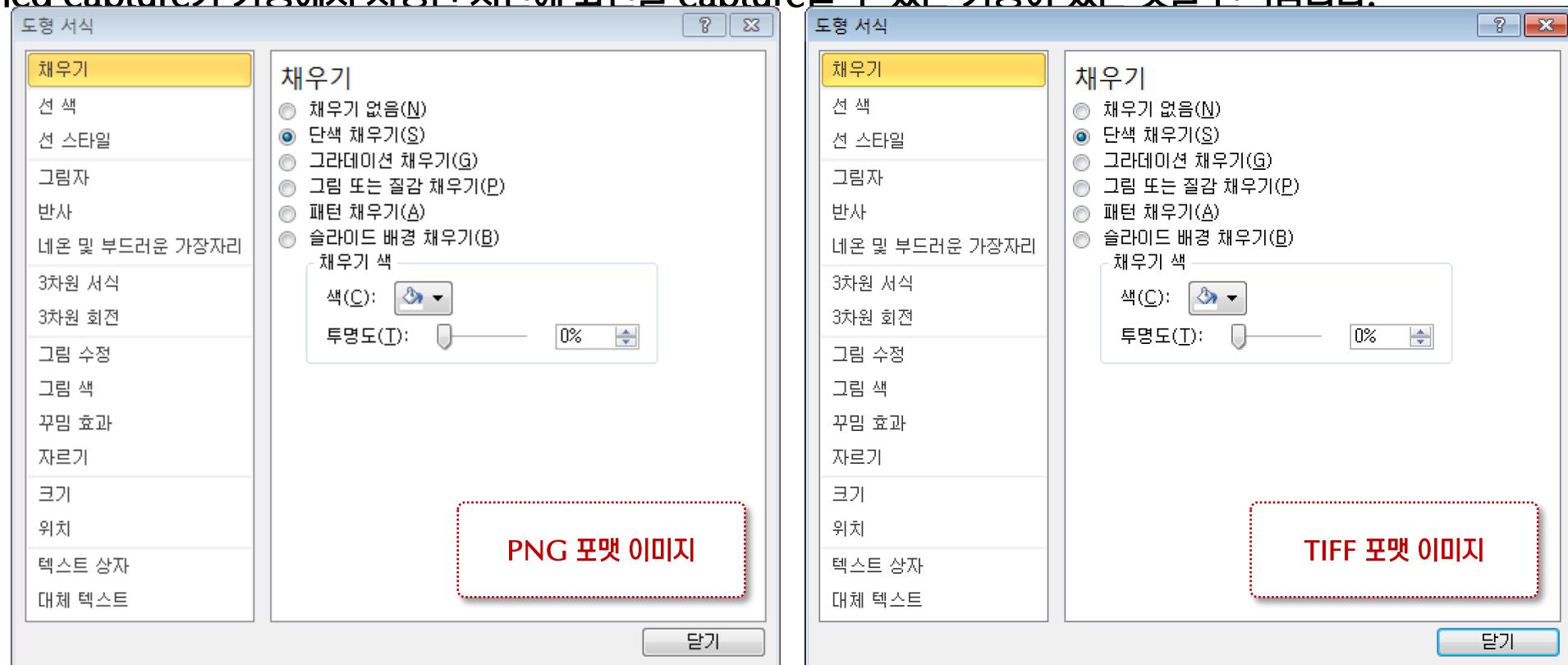


캡쳐도구 예

- ✓ 애플 Grab 애플리케이션 → Timed Screen 기능
- ✓ 애플 Grab 애플리케이션 → 고화질 이미지 캡쳐

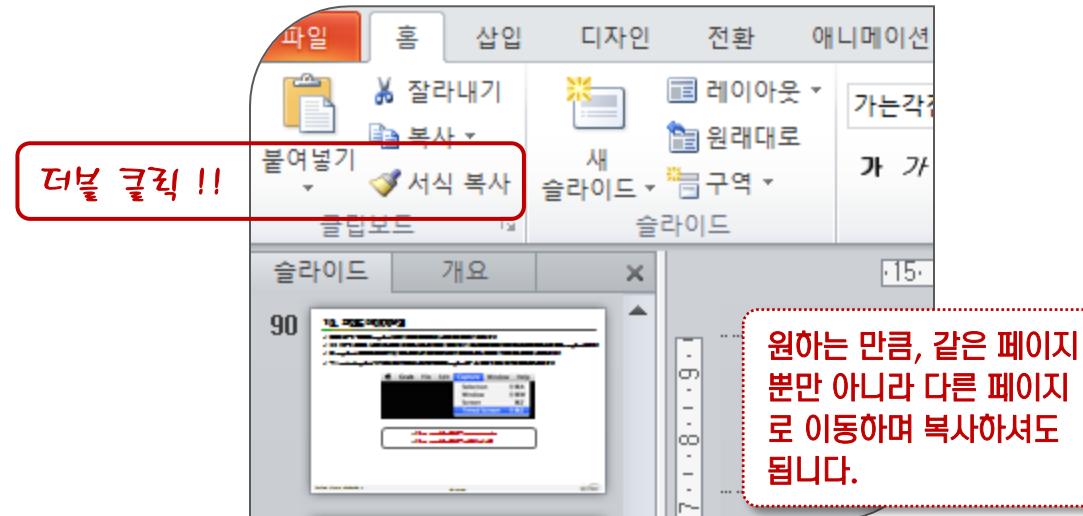
## 9.10 고화질 이미지(2/2)

- ✓ 컴퓨터의 화면을 capture하여 자료를 만드는데 사용하는 경우가 많습니다.
- ✓ 이 때의 문제는 낮은 해상도와 (마우스 커서가 눌려 있을 때나 대상 위에 있을 때만 동작하는) 컨텍스트 메뉴 capture입니다
- ✓ Capture 도구를 선택할 때, 압축률은 낮지만 고화질을 지원하는 tiff 포맷을 지원하는 도구를 선택합니다.
- ✓ Timed capture가 가능해서 지정된 시간에 화면을 capture할 수 있는 기능이 있는 것을 선택합니다.



## 9.11 서식 복사

- ✓ 서식 복사 메뉴를 클릭하면 현재 선택한 도형의 서식을 복사하며, 대상을 클릭하면 서식이 복사됩니다.
- ✓ 그런데, 복사 대상이 여러 곳일 때 이 기능은 지속적인 반복을 해야 합니다.
- ✓ 이럴 때, “서식 복사” 메뉴를 더블 클릭하여 선택하면, 여러 대상으로 서식 복사를 할 수 있습니다.



## 9.12 텍스트 상자 음영

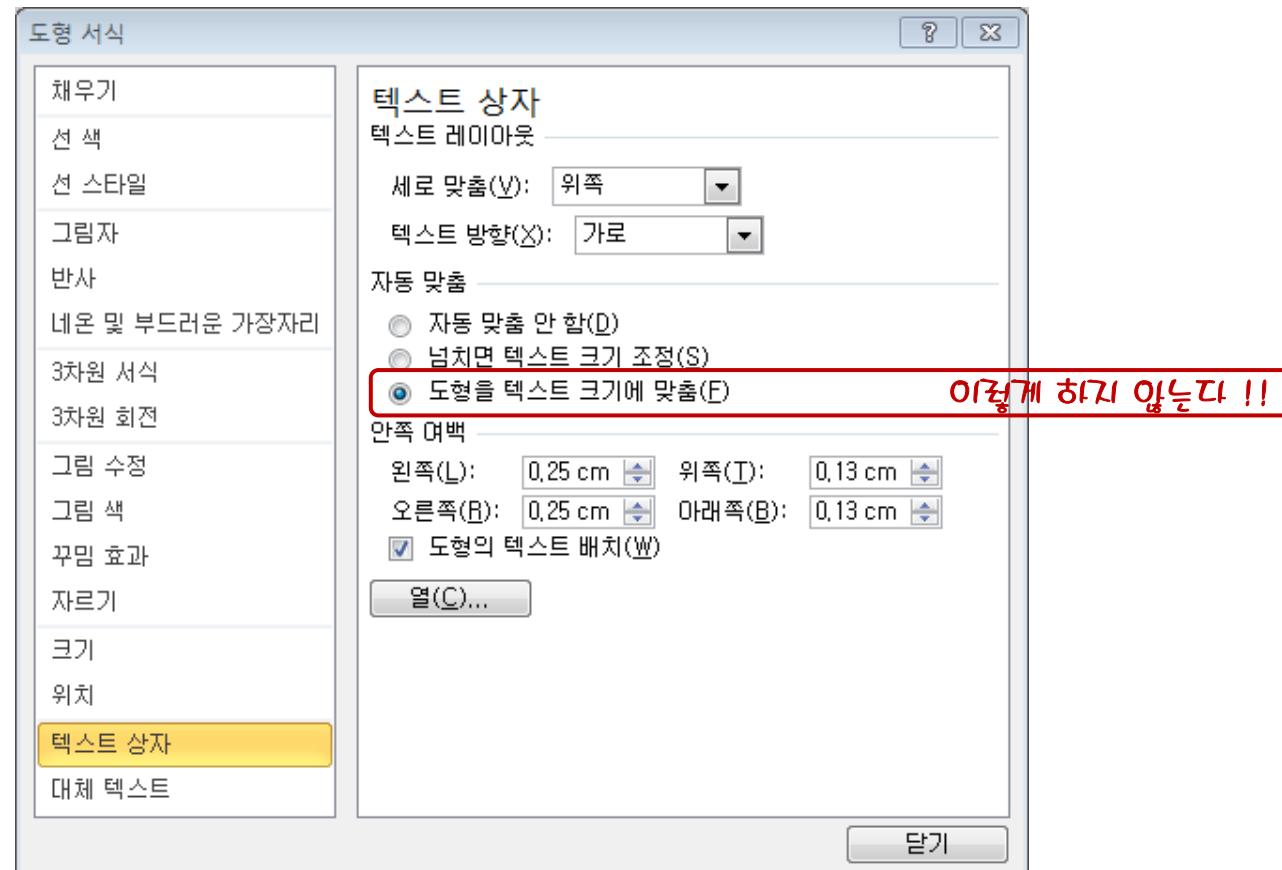
- ✓ 도형에 텍스트가 있을 경우, 텍스트 상자의 주변에 그림자 처리를 하는 경우가 있습니다. 아래 그림 참조
- ✓ 이런 도형을 포함하고 있는 파일을 PDF로 저장할 경우, 텍스트 상자는 이미지로 저장됩니다.
- ✓ **이 경우 문제는, 텍스트를 명확하게 보여 주기 위해 높은 해상도를 가진 이미지로 저장됩니다.**
- ✓ 이런 도형이 많을 경우, PDF 파일 크기가 커집니다. 5MB 크기의 PPT 파일이 18MB 크기의 PDF 파일이 되기도합니다.

```
public List<User> readAllUsers() {  
    Connection conn = null;  
    PreparedStatement pstmt = null;  
    ResultSet rs = null;  
    List<User> users = new ArrayList<User>();  
  
    try {  
        conn = dataSource.getConnection();  
        StringBuffer sb = new StringBuffer(512);  
        sb.append("SELECT a.user_id, a.user_nm, a.email, a.password ");  
        sb.append(" FROM user_tb a");  
        pstmt = conn.prepareStatement(sb.toString());  
  
        rs = pstmt.executeQuery();  
  
        while (rs.next()) {  
            users.add(mapToUser(rs));  
        }  
        ...  
    }  
}
```

음영 처리 주의 !!

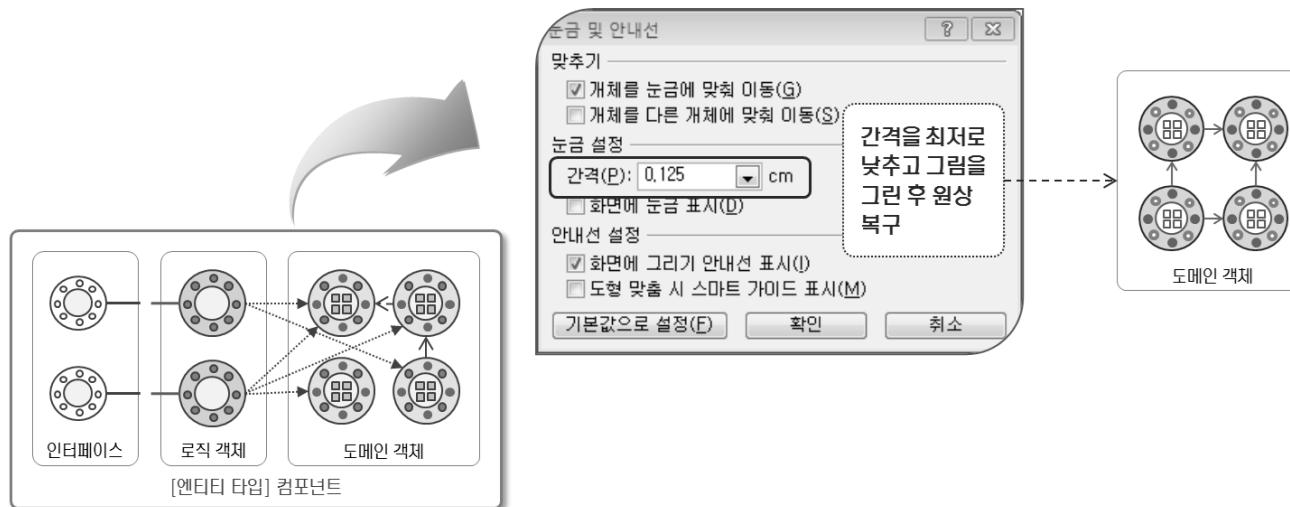
## 9.13 도형을 텍스트 크기에 맞춤

- ✓ 텍스트를 포함하고 있는 도형의 경우, 주변의 도형과 크기를 맞추어야 하는 경우가 많습니다.
- ✓ 그런데 도형의 크기가 조절되지 않는 경우가 있습니다. 그 결과 주변과 크기가 맞지 않습니다.
- ✓ “도형을 텍스트 크기에 맞춤”을 체크 했기 때문입니다.
- ✓ “자동 맞춤 안 함”으로 선택하여 자유롭게 도형의 크기를 조절 할 수 있습니다.



## 9.14 요약

- ✓ 열 가지 유용한 Tip을 서술했습니다. 어떤 것은 잘 드러나지 않는 메뉴이고, 어떤 것은 경험의 산물입니다.
- ✓ Tip이란 알고 있을 때, 많은 시간과 노력을 절약해 주는 경우도 있습니다.
- ✓ 특히, 페이지 복사 후 레이아웃 조정 같은 기능은 모르고 있을 경우, 파일 간 페이지 복사가 무한 반복 수작업이 됩니다.
- ✓ 프리젠테이션 도구의 버전에 따라 메뉴의 위치나 기능이 서로 다른 경우가 있으므로 주의합니다.



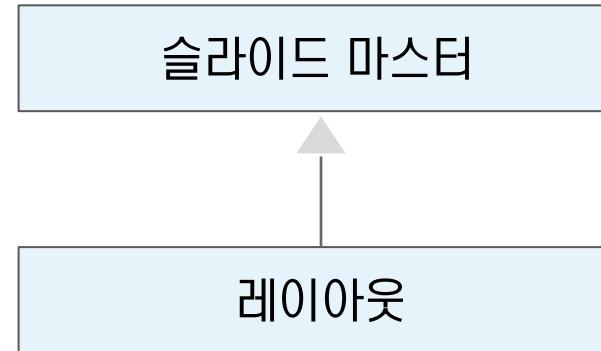


## 10. 템플릿

- 
- 10.1 슬라이드 마스터
  - 10.2 레이아웃
  - 10.3 목차 레이아웃
  - 10.4 요약

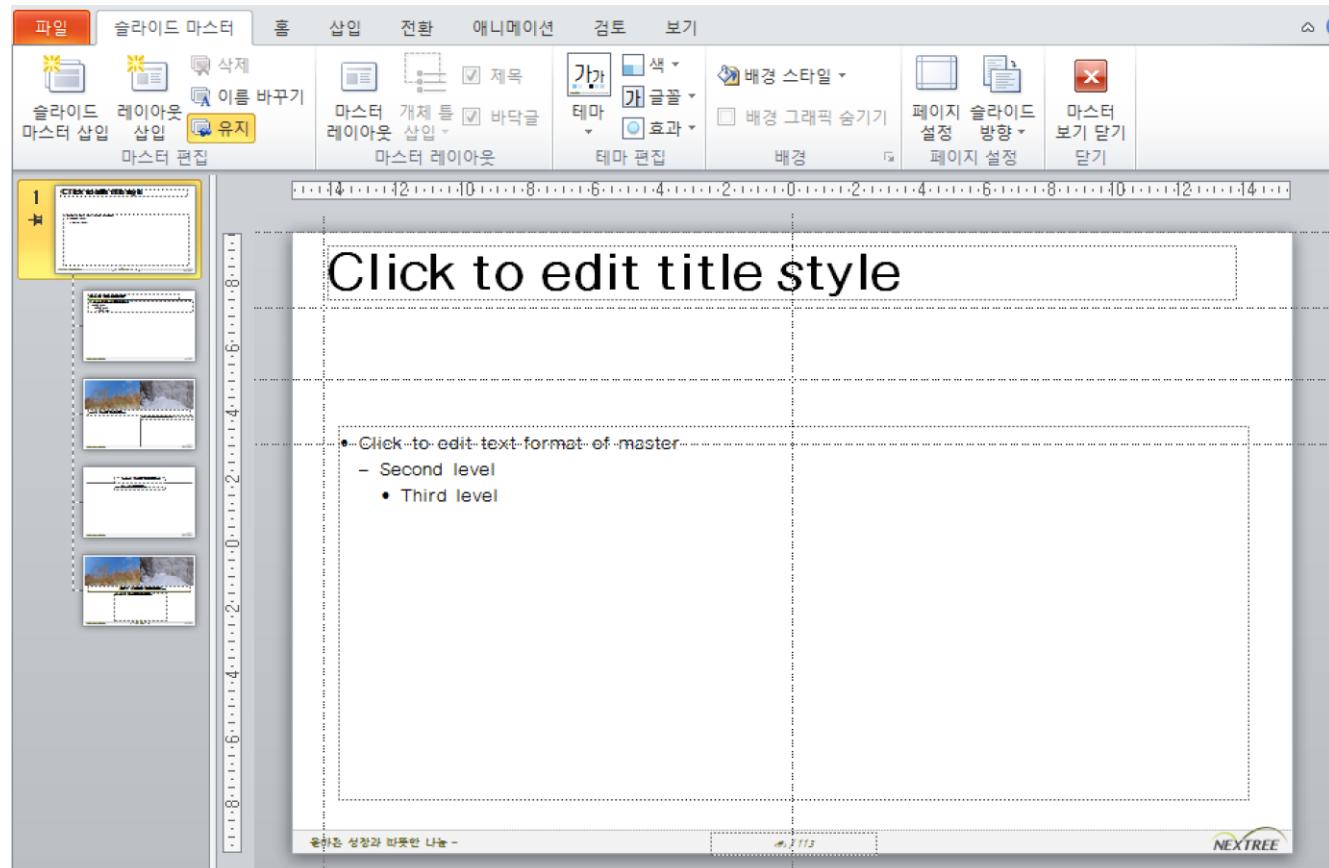
## 10.1 슬라이드 마스터(1/2)

- ✓ 슬라이드 마스터와 레이아웃은 상속(inheritance) 관계를 가집니다. 슬라이드 마스터의 서식을 레이아웃이 이어 받습니다.
- ✓ 페이지가 만들어지기 전이라면 바로 상속이 됩니다.
- ✓ 레이아웃을 만들고 페이지가 만들어 진 후라면 주요 개체(페이지 번호 등)을 제외하고는 상속 받지 않습니다.
- ✓ 따라서, 슬라이드 마스터를 신중하게 설계한 후에 레이아웃을 만들어 갑니다.



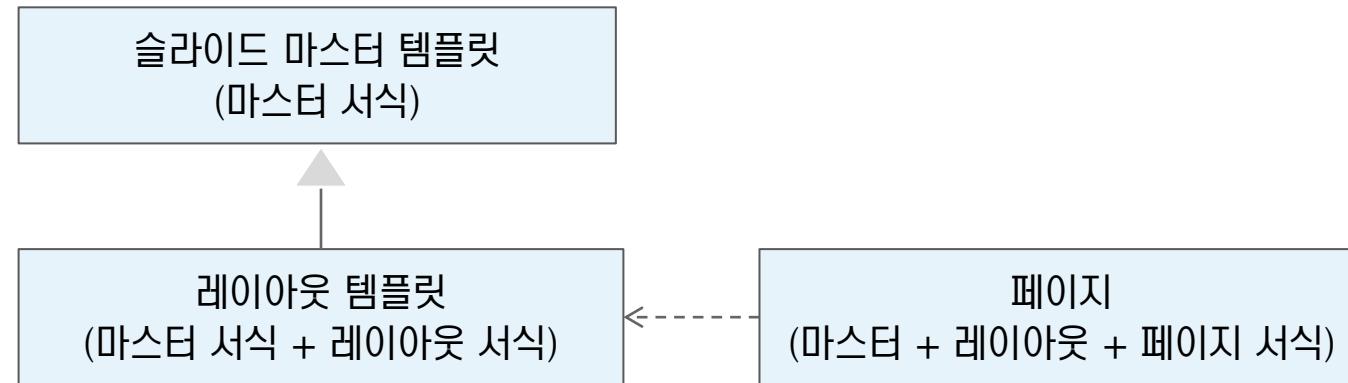
## 10.1 슬라이드 마스터(2/2)

- ✓ 슬라이드 마스터는 모든 레이아웃에 들어가야 할 개체와 서식을 담고 있습니다.
- ✓ 페이지, 목차, 표지 등에 공통으로 들어갈 것이 없는 경우에 그냥 아무 서식을 지정하지 않습니다.
- ✓ 아래 예제는 바닥글을 정의했습니다. 이후 만드는 모든 레이아웃 템플릿에는 동일한 바닥글이 들어갑니다.
- ✓ 이미 레이아웃이 있고, 그 레이아웃을 사용하는 페이지가 있는 경우에, 추가되는 서식은 대부분 반영되지 않습니다.



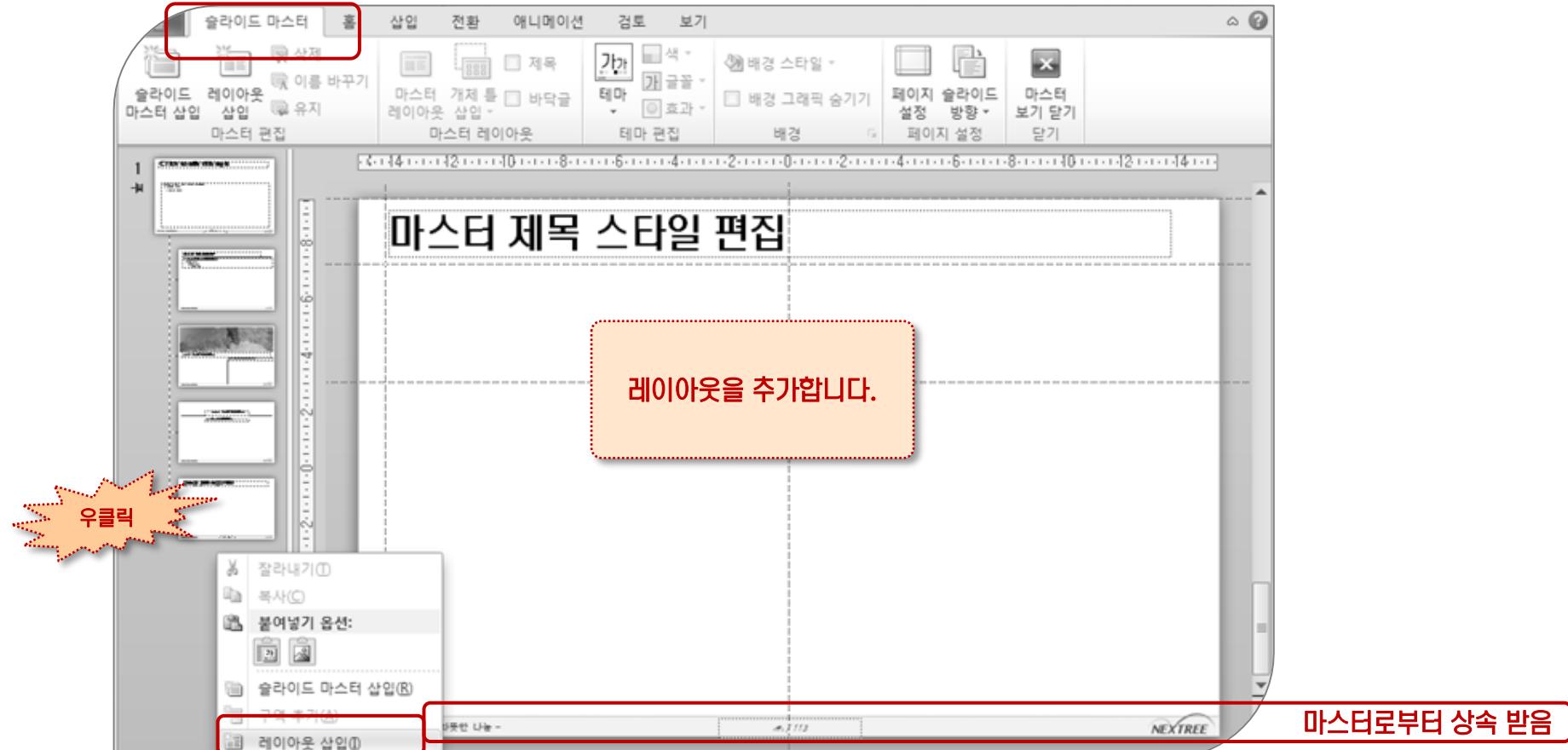
## 10.2 레이아웃

- ✓ 페이지는 레이아웃 템플릿을 틀로 하여 만듭니다. 초기 서식은 레이아웃의 서식을 그대로 가져옵니다.
- ✓ 레이아웃에서 지정된 서식이라도 페이지에서 변경할 수 있습니다.
- ✓ 페이지에서 변경하면 이 서식은 이제 페이지 서식이 되므로, 레이아웃에서 조정할 수 없게 됩니다.
- ✓ 레이아웃의 서식(타이틀, 개체틀, 바닥글)은 가급적 그대로 두는 것이 좋습니다. 필요할 경우, 전체 조정이 가능합니다.



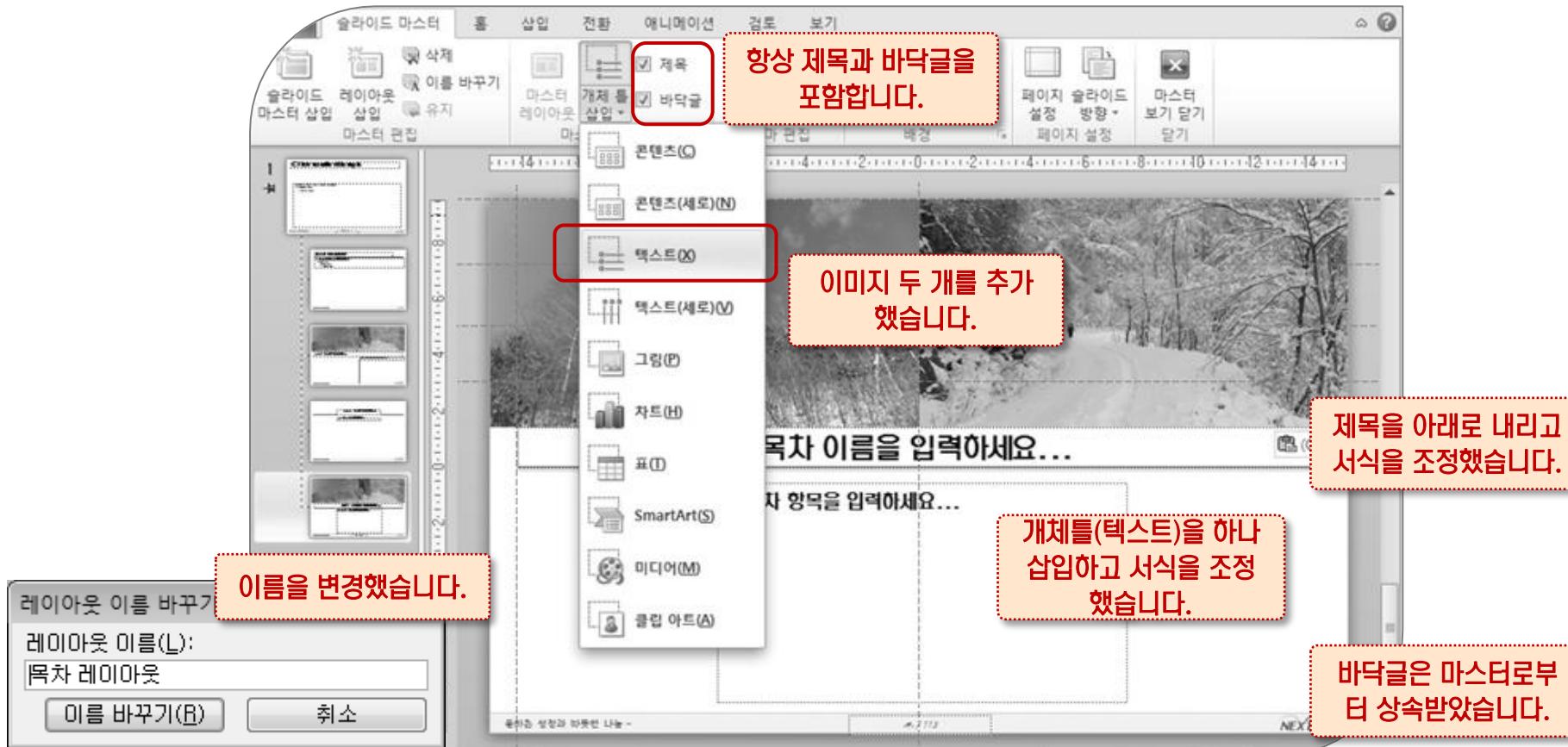
### 10.3 목차 레이아웃(1/3) – 추가

- ✓ 레이아웃을 추가하는 간단한 작업을 해 봅니다.
  - ✓ 생성된 레이아웃의 바닥글이 슬라이드 마스터로부터 서식을 상속 받음을 확인할 수 있습니다.
  - ✓ 레이아웃은 필요한 최소한으로 보유하는 것이 좋습니다. 너무 많은 레이아웃은 혼란스럽기만 합니다.



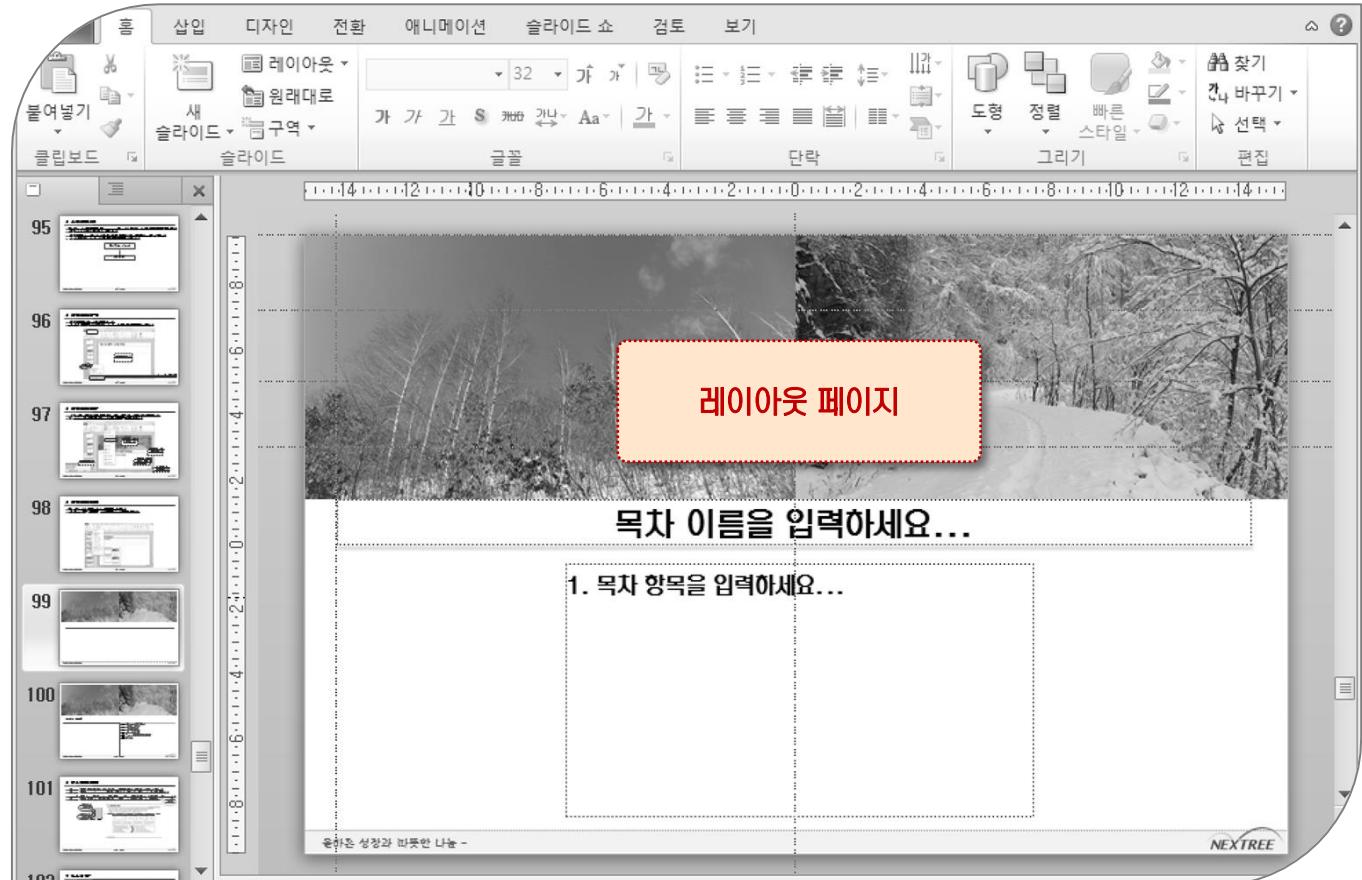
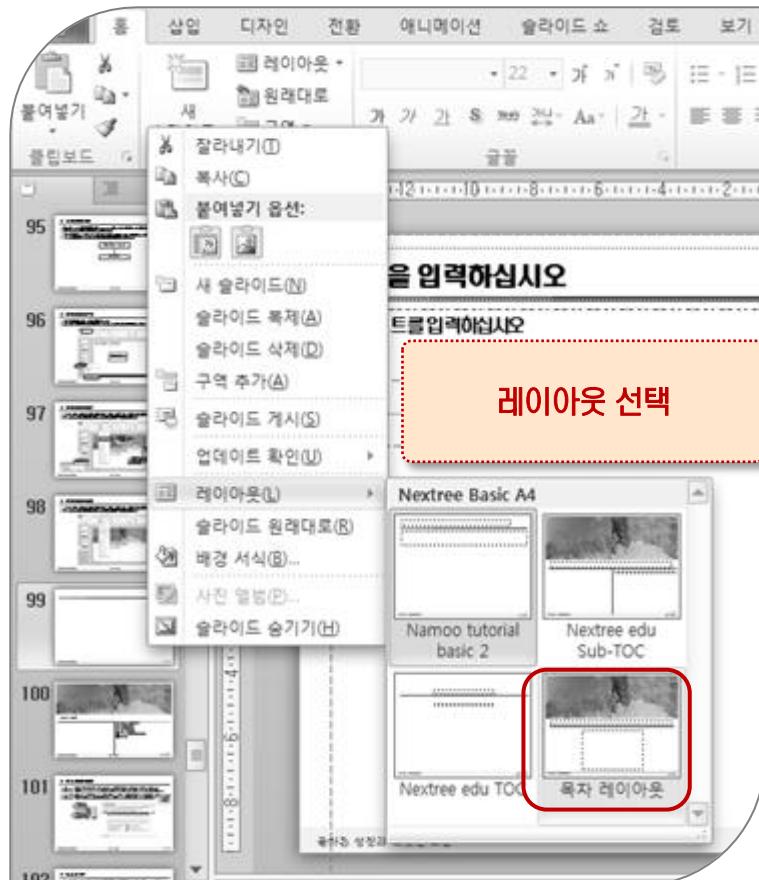
## 10.3 목차 레이아웃(2/3) – 편집

- ✓ 타이틀을 중간으로 이동하여 서식을 편집하고, 안내문으로 “목차 이름을 입력하세요...”를 입력합니다.
- ✓ 개체를 삽입 → 텍스트 하여 목차 항목을 추가합니다. 안내문으로 “목차 항목을 입력하세요...”를 입력합니다.
- ✓ 이미지를 가져다가 레이아웃의 윗부분에 두고 크기를 조정합니다.
- ✓ 이미지를 가져다가 윗쪽을 장식하고, 마지막으로 레이아웃의 이름을 변경합니다.



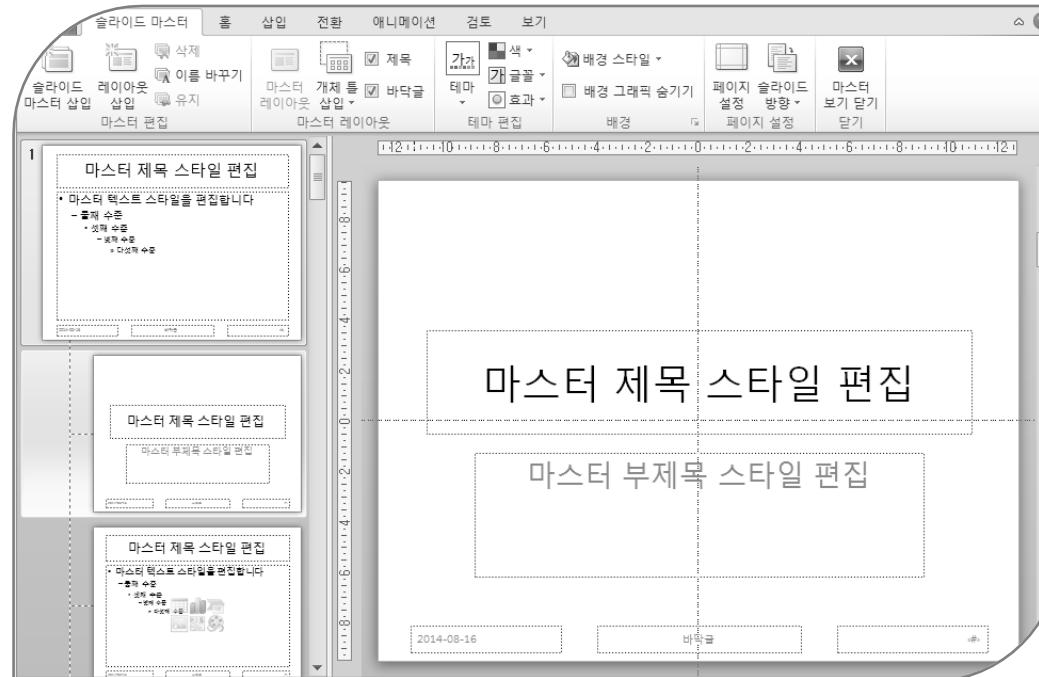
### 10.3 목차 레이아웃(3/3) – 사용

- ✓ 슬라이드를 하나 추가합니다. 기본 레이아웃으로 설정됩니다. 레이아웃 → 목차 레이아웃(방금 만들었던)을 선택합니다.
  - ✓ 새로 만든 레이아웃 페이지가 만들어 졌습니다. 이제 목차 이름을 입력하고, 항목을 채워서 목차를 만드시면 됩니다.
  - ✓ 일단 페이지가 만들어 지면, 이 레이아웃은 삭제할 수 없습니다.
  - ✓ 개체의 서식을 변경하지 않는다면 레이아웃에서 한꺼번에 서식을 변경할 수 있습니다.



## 10.4 요약

- ✓ 템플릿 작업이 어려운 이유는 슬라이드 마스터 – 레이아웃 템플릿 – 페이지 사이에 설정되어 있는 규칙 때문입니다.
- ✓ 세 가지 요소 간에 걸려 있는 규칙을 이해하면, 템플릿 작업은 매우 간단합니다.
- ✓ 어떤 경우든, 템플릿을 미세 조정하는 작업은 많은 시간과 노력을 요구합니다.
- ✓ 잘 정의된 템플릿을 공유하면, 자료 만드는 시간을 절약할 수 있습니다.





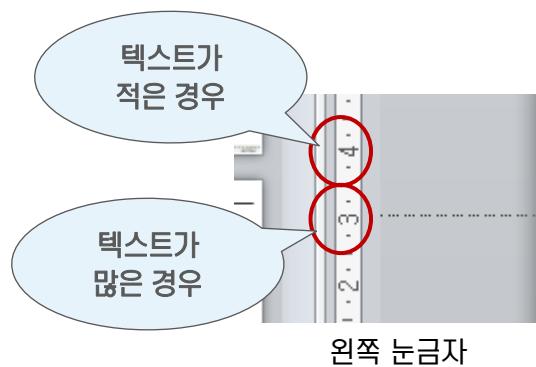
## 11. 문장

- 
- 11.1 텍스트 위치와 크기
  - 11.2 접속사 “및”
  - 11.3 서술어 오류
  - 11.4 명사형 나열
  - 11.5 ‘~대하여’ 혹은 ‘~관하여’
  - 11.6 요약

# 11.1 텍스트 위치와 크기

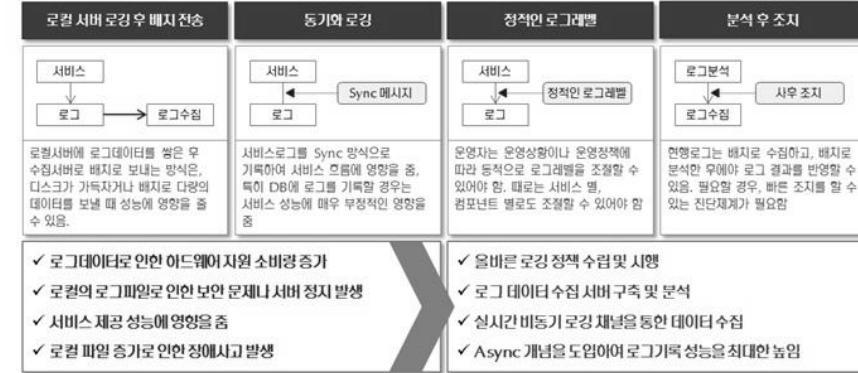
- ✓ 텍스트 영역은 대체로 위 쪽입니다. 아래나 좌측 또는 우측에 배치하는 경우도 있지만 흔하지는 않습니다.
- ✓ 텍스트 영역의 폰트 크기는 20, 18, 16 포인트 중에 하나를 선택합니다. 선택한 크기는 전 문서에 적용합니다.
- ✓ 보고서 성격의 자료는 16 포인트, 발표 자료일 경우는 20 포인트, 컨텐츠 정리와 발표 자료일 경우 18포인트를 씁니다.
- ✓ 텍스트 영역의 배경은 기본 배경색을 사용하며, 폰트 유형, 크기 등에 대한 변화는 가급적 없어야 합니다.

18 포인트로  
작성한 텍스트



## 2. 스마트로깅 개요

- ✓ 로깅 구조는 수행성능, 스토리지 자원, 보안, 장애, 운영분석 등에 영향을 주는 매우 중요한 아키텍처 요소임
- ✓ 현행 로깅 구조는 성능, 자원, 보안, 분석 등의 측면에서 스마트하지 못한 구조를 가지고 있음
- ✓ 올바른 로깅 구조 설계 및 기술적용을 통해 스마트 로깅 목표를 달성하여야 함



올바른 성장과 따뜻한 나눔...

1-4

## 11.2 접속사 ‘및’

- ✓ '～와(과)' 또는 '～하고'라고 하면 될 문장에 '및'을 여러 개 중첩하여 매끄럽지 못한 문장을 쓰는 실수를 합니다.
- ✓ '및'을 쓰지 않고도 깔끔하면서 잘 읽히는 문장을 쓸 수 있습니다.

(나쁜 예) 타부서 및 타기관의 요청에 대하여 신속 및 정확한 대응 및 방안을 제시한다.

(좋은 예) 타부서와 타기관의 요청에 신속하고 정확하게 대응하고 방안을 제시한다.

문장을 깔끔하게 쓰기 위한 9가지 팁 (<http://www.infuture.kr/m/post/338>) , 유정식 님의 블로그에서...

## 11.3 서술어 오류 [1/2]

- ✓ 영어로 인해 잘못된 서술어 오류가 많습니다.
- ✓ '～하도록 한다'라는 서술어는 누군가(타인)에게 무언가를 하도록 만들겠다는 뜻으로, 사역동사(make 등)를 쓰는 영어에서는 가능한 표현이지만 국어에서는 거북한 표현입니다.

(나쁜 예) 마케팅 계획 및 전략 수립시 OOO부서의 입장을 반영하도록 한다.

(좋은 예) 마케팅 계획과 전략을 수립할 때 OOO부서의 입장을 반영한다.

(나쁜 예) OO을 제안하여 XX시스템을 변경시킨다.

(좋은 예) OO을 제안하여 XX시스템을 변경한다.

- ✓ 영어 번역 문장에 길들여져 'can'이나 'may'에 해당하는 '～할 수 있다'라는 서술어를 지나치게 많이 사용하는 경향이 있습니다.

(나쁜 예) 적절한 담당자의 도움을 받아 연구 및 프로젝트 수행을 할 수 있다.

(좋은 예) 적절한 담당자의 도움을 받아 연구와 프로젝트를 수행한다.

문장을 깔끔하게 쓰기 위한 9가지 팁 (<http://www.infuture.kr/m/post/338>) , 유정식 님의 블로그에서...

## 11.3 서술어 오류 [2/2]

- ✓ 군더더기 문장이 많습니다.
- ✓ 동작이 계속되는 상황을 표현하는 '～하고 있다'라는 서술어가 많이 쓰입니다. 그러나 문장에 자신이 없기 때문에 '～하고 있다'가 남발됩니다. 진정한 문장력은 짧게 쓰는 용기에서 나옵니다.

(나쁜 예) OO산업 및 XX시스템에 대해 이해하고 있다.

(좋은 예) OO산업과 XX시스템을 이해한다.

문장을 깔끔하게 쓰기 위한 9가지 팁 (<http://www.infuture.kr/m/post/338>) , 유정식 님의 블로그에서...

## 11.4 명사형 나열

- ✓ 명사형 단어들을 지나치게 쓰면, 가독성과 문장에 대한 이해를 떨어뜨립니다.
- ✓ 명사형을 지양하고 서술어를 적절하게 사용하여야 합니다.

(나쁜 예) 프로젝트 진행 과정 판단 미숙으로 문제 발생 확률 예측 실패 야기 가능성을 점검한다.

(좋은 예) 프로젝트 진행을 잘못 판단하여 문제가 발생할 확률을 예측하지 못하는지 점검한다.

문장을 깔끔하게 쓰기 위한 9가지 팁 (<http://www.infuture.kr/m/post/338>) , 유정식 님의 블로그에서...

## 11.5 '~대하여' 혹은 '~관하여'

- ✓ '~대하여'라는 문구가 들어가면 뭔가 대단한 내용을 이야기하듯 느껴지는 효과가 있기 때문에 공식적인 문서에서 많이 사용합니다.
- ✓ 하지만 지나치게 사용하면 어색한 문장이 되고, '~대하여' 혹은 '~관하여'라는 말을 쓰지 않고도 얼마든지 간결하게 기술이 가능합니다.

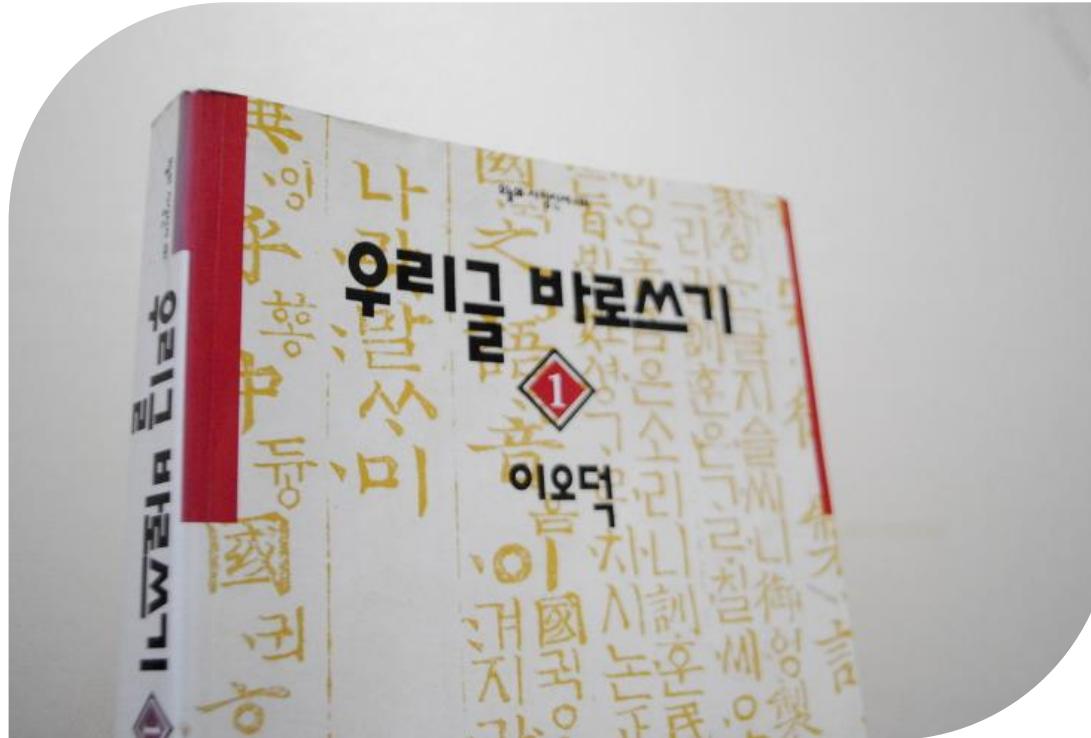
(나쁜 예) OO분석 결과에 대하여 문제점을 발견하고 신규제품 지식에 관하여 숙지할 수 있다.

(좋은 예) OO분석 결과에서 문제점을 발견하고 신규제품 지식을 숙지한다.

문장을 깔끔하게 쓰기 위한 9가지 팁 (<http://www.infuture.kr/m/post/338>) , 유정식 님의 블로그에서...

## 11.6 요약

- ✓ 좋은 문장을 위한 책, 이오덕 선생의 “우리글 바로 쓰기”를 추천합니다.





## 12. 전달(presentation)

- 12.1 이야기
- 12.2 자세
- 12.3 포인팅
- 12.4 시선
- 12.5 위치
- 12.6 시간
- 12.7 열정
- 12.8 유머
- 12.9 발표 기회

# 이야기

# 자세

# 포인팅

# 시선

# 위치

# 시간

# 열정

# 유머

# 발표 기회

- ✓ 질문과 대답
- ✓ 토의

감사합니다...

- ❖ 송태국 (tsong@nextree.co.kr)
- ❖ 넥스트리컨설팅(주)
- ❖ [www.nextree.co.kr](http://www.nextree.co.kr)