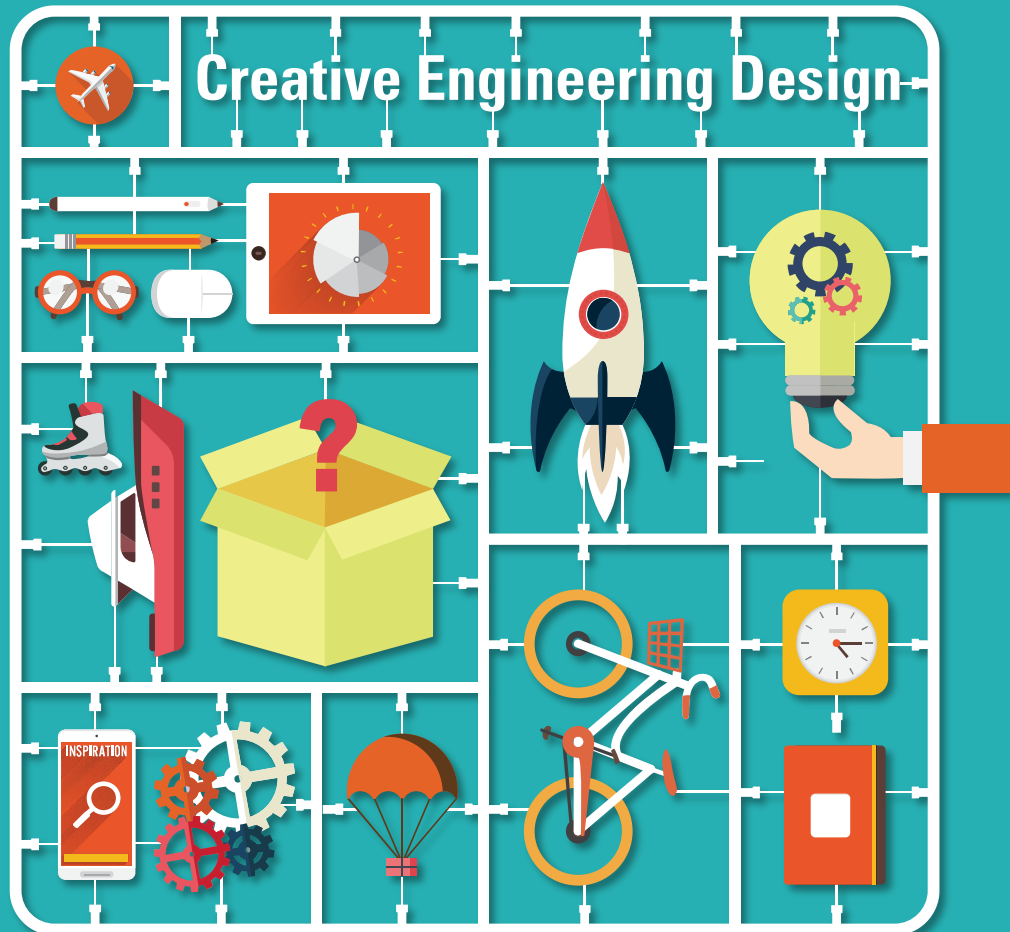


“본 강의 동영상 및 자료는 대한민국 저작권법을 준수합니다. 본 강의 동영상 및 자료는 상명대학교 재학생들의 수업목적으로 제작·배포되는 것이므로, 수업목적으로 내려받은 강의 동영상 및 자료는 수업목적 이외에 다른 용도로 사용할 수 없으며, 다른 장소 및 타인에게 복제, 전송하여 공유할 수 없습니다. 이를 위반해서 발생하는 모든 법적 책임은 행위 주체인 본인에게 있습니다.”



# 창의공학설계를 위한 단계 분석

Introduction to Engineering Design

공학설계입문

# 이번 주 강의 개요

이번 시간에는 이전에 다룬 창의적 공학설계를 위한 7단계 프로세스를 단계별로 심화 학습합니다.

- 공학설계 프로세스와 문제 정의
- 정보 수집
- 해결책 생성 등의 단계들



# 공학설계 프로세스와 문제 정의

# 공학설계 7단계 프로세스

- [1단계] 문제 정의(Define the problem)
- [2단계] 정보 수집(Collect information)
- [3단계] 해결책 생성(Generate multiple solutions)
- [4단계] 분석과 선택(Analyze and select a solution)
- [5단계] 프로토타입 만들기(Build a prototype)
- [6단계] 테스트와 성능 개선(Test and improvement)
- [7단계] 설계의 구현(Implement the design)

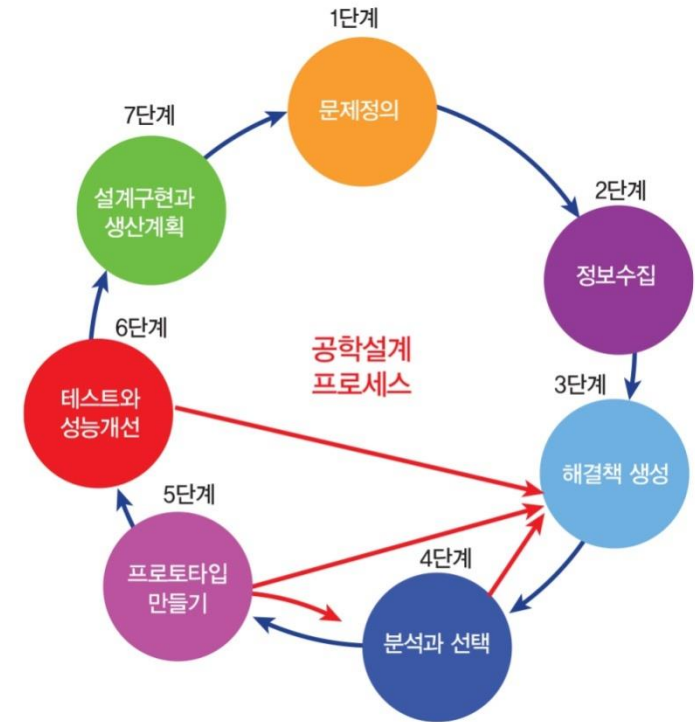


그림 4.8 7단계 공학설계

# 문제 정의 단계

- 공학설계 프로세스의 첫 단계로 큰 중요성을 가짐
  - 애매하거나 잘못된 문제 정의 → 연구 개발을 열심히 해도 잘못된 결과로 이어질 수 있음
  - 문제를 구체적이고 자세하게 명시해야 함
- 엔지니어는 수요자 또는 의뢰자의 요구사항을 정확히 파악할 필요가 있음
  - B2B일 경우 의뢰자, B2C일 경우 수요자(소비자)
- 올바른 문제 정의를 위해서는, 충분한 사전 연구가 필요한 경우도 있음
  - 이전 제품에 대해 소비자로부터 불만이 있었던 부분은 무엇인지?
  - 소비자가 신제품에서 원하는 기능은 무엇인지?
  - 현재 시장 트렌드는 어떤지?

What's the Problem?



# 공학적 문제

- 공학적 문제는 대부분이 개방형 문제이거나 제대로 정의되지 않은 경우가 많음
  - 문제 자체가 애매하고, 이로 인해 설명이 부족한 경우가 존재
  - 명확한 제한조건, 요구사항이나 해결법이 존재하지 않는 경우도 있음
- 열린 사고력 문제 (개방형 문제, open-ended problem)
  - 같은 문제에 대해서 다양한 해결 방안을 가지는 문제
  - 예) 10만명의 동시접속자를 감당할 수 있는 게임서버는 어떻게 설계해야 하는가?
- 비정형 문제 (ill-structured problem)
  - 해답이 일정하지 않은 문제
  - 예) 새로 나오는 신제품의 색상은 어떤 것이 적절한가?
- 문제의 본질을 파악하여, 진짜 문제(real problem)를 정의할 필요가 있음
  - 진짜 문제에 관한 내용은 다음주에 다룹니다

What's the Problem?



# 문제 정의를 위한 기본적인 질문들

- 이 연구개발의 목적은 무엇인가?
- 문제의 배경은 무엇인가?
- 추가적인 배경 지식이 필요한가?
- 결과의 성공 여부를 판단할 기준은 무엇인가?
- 어떤 기능을 포함해야 하는가?
- 외부의 도움이 필요한가?
- 비용은 얼마나 들 것인가?
- 걸리는 시간은 적절한가?
- 언제부터 시작할 것인가?

# 엑시노스 2200에 탑재된 AMD mRDNA2로 살펴본 문제 정의

- 2019년 삼성전자와 AMD는 초저전력/고성능 그래픽 설계자산(IP)에 대한 전략적 파트너십 체결
- 2022년 엑시노스 2200에 AMD와 삼성이 함께 설계한 mRDNA2 GPU가 포함될 예정
  - 기존 데스크탑용의 RDNA2 아키텍처를 모바일용으로 커스터마이징
  - 갤럭시 S22에 탑재될 예정
- 삼성은 왜 AMD와 함께 새로운 GPU IP를 직접 개발하려고 했을까요?





# 문제 정의를 위한 기본적인 질문들 – mRDNA2 사례

- 이 연구개발의 목적은 무엇인가? 스마트폰에 들어갈 새로운 GPU IP 설계
- 문제의 배경은 무엇인가? 기존 ARM 말리 GPU가 들어간 엑시노스의 그래픽 성능/발열 문제
- 추가적인 배경 지식이 필요한가? GPU 설계 및 구현에 대한 노하우 필요
- 결과의 성공 여부를 판단할 기준은 무엇인가? 더 높은 성능, 더 좋은 화질, 더 낮은 발열
- 어떤 기능을 포함해야 하는가? Variable-rate shading, ray tracing 등
- 외부의 도움이 필요한가? 자체 개발로는 경쟁이 힘들므로, 전문 GPU 회사와의 협업 필요
- 비용은 얼마나 들 것인가? (아마도 몇천억원?)
- 걸리는 시간은 적절한가? 2019년에 협약을 맺은 이후, 결과적으로 2022년에 제품 출시
- 언제부터 시작할 것인가? 협약 직후부터 공동 연구개발 시작

# 5단계 문제 정의 프로세스

- **[1단계] 명확한 목표 설정**
  - 이 설계의 목표는 무엇인가?
  - 구체적으로 무엇을 이루고 싶은가?
  - 어떤 방법으로 목표를 달성할 것인가?
  - 기타 연구개발과 관련된 질문 서술
- **[2단계] 문제의 배경 파악**
  - 문제 해결 동기 명시
  - 문제와 관련된 배경 지식 및 증상 파악
  - 기존의 해결방법들을 조사
- **[3단계] 제한 조건 파악**
  - 문제에서 파악해야 할 영역들을 명시
  - 제약조건 및 장애요인을 명확히 기술
  - 구현될 시스템의 한계점 명시 (작동 수준, 성능 등)
- **[4단계] 주요 기능 설정**
  - 목표 리스트로부터 수행되어야 할 구체적 기능 설정
  - 세부 블록에서 요구되는 기능들을 체계적으로 정립
  - 반드시 구현되어야 할 기능들과 타협 가능한 기능들을 분리
  - 성공적인 수행 판단의 기준 설정
- **[5단계] 요구사항 설정**
  - 문제 해결에 필요한 기능적, 외형적, 능력적 요구사항들을 구체적으로 설정



그림 6.2 5단계 문제 정의 프로세스

# 공학설계에 있어서의 기능적, 비기능적 요구사항

- 기능적 요구사항 (Functional Requirements)
  - 시스템의 가지고 있는 기능과 관련된 요구사항

UC_02.1	지출 내역을 생성한다
설명	사용자는 모임의 결제 내역을 추가할 수 있다
행위자	사용자
선행조건	없음
후행조건	결제 내역이 생성된다.
기본 동작	<ol style="list-style-type: none"> <li>1. 지출 내역 생성 버튼을 누른다</li> <li>2. 결제내역과 금액을 입력할 수 있는 다이얼로그가 생성된다</li> <li>3. 결제 내역과 금액을 입력한다.               <ul style="list-style-type: none"> <li>✓ 금액은 최대 1억원 까지만 입력 받을 수 있도록 한다.</li> <li>✓ 금액은 0원 이상의 정수만 받도록 한다</li> </ul> </li> <li>4. 결제 참가자를 고를 수 있는 다이얼로그가 생성된다</li> <li>5. 결제 참가자를 선택한 후 확인 버튼을 누른다</li> <li>6. 결제 내역이 생성된다</li> <li>7. 대신 결제한 사람을 고르는 다이얼로그가 생성된다</li> <li>8. 있는 경우는 참가자를 고르고 확인 버튼을, 없는 경우에는 취소 버튼을 누른다</li> <li>9. 지출 내역이 생성된다</li> </ol>
추가 동작	없음

Requirements



아는 개발자 :: 기능적/비기능적 요구사항 - 구조설계서 작성 (1/4) ([selfish-developer.com](http://selfish-developer.com))

# 공학설계에 있어서의 기능적, 비기능적 요구사항

- 비기능적 요구사항 (Non-Functional Requirements)
  - 기능적 요구사항에서 다루지 못한 품질적인 요소와 관련된 요구사항

NFR_01	정확성	시스템이 계산한 금액 정보는 정확해야 한다.
설명	시스템이 제공하는 서비스를 이용하는 사용자의 입장에서 정산된 금액은 실제 정산 금액 값과 동일해야 한다.	
환경	사용자가 시스템을 이용해 모임 비용을 정산한다	
자극	없음	
반응	없음	
측정	시스템에서 정산한 금액과 수기로 정산한 금액이 동일한지 확인한다	
제약	[시스템 정산 금액] = [수기 정산 금액]	

Requirements



[아는 개발자 :: 기능적/비기능적 요구사항 - 구조설계서 작성 \(1/4\) \(selfish-developer.com\)](http://selfish-developer.com)

# 절대적 규격과 타협 가능한 규격

- 규격에 포함 가능한 예시
  - 작동(속도, 문제 해결)
  - 기능(입력, 출력, 작동 모드)
  - 물리적 성질(크기, 무게, 온도)
  - 사용의 용이성
  - 신뢰성, 내구성, 보안성
  - 파워(전압 수준, 배터리 수명)
  - 적용 표준에 부합
  - 기존의 제품들과 호환이 가능
  - 비용 등
- 절대적 규격: 반드시 지켜야 할 규격
- 절충 가능한 규격: 어느 정도 범위 내에서는 받아들여질 수 있는 규격



# 정보 수집

# 정보 수집을 위한 기본적인 질문들

- 어떤 정보들이 필요하고 어떤 처리 과정을 거칠 것인가?
- 누가 얼마동안 자료를 수집해야 할까?
- 해결법이 이미 존재한다면 기존의 해결법들은 무엇인가?
- 새로운 해결 방법은 정말로 필요한가? 그렇다면, 기존 해결법에서 무엇을 개선해야 할까?
- 해결법에서 중요한 요소는 무엇인가? (안전성, 심미성, 환경, 경제성 등)

적절한 정보의 수집은 정의된 문제를 다시 한번 검토함으로써, 오류나 잘못된 설정을 발견 가능

# 자료 수집 방법

- 인터넷 – 네이버, 구글 등의 검색 엔진 사용
- 인터뷰 – 사용자 또는 전문가
- 관련 서적, 논문, 특허, 등록 상표, 연구보고서, 상품 카탈로그 등



그림 6.3 인터넷과 인터뷰를 통한 자료 수집



그림 6.4 특허와 등록상표



# 정보 수집을 위한 자료의 종류

- 1차 자료 (primary data)
  - 연구자가 설문이나 면접 등을 통해 직접 수집, 작성한 자료
  - 조사목적에 적합한 자료를 수집할 수 있으나, 많은 시간과 비용이 필요
- 2차 자료 (secondary data)
  - 1차 자료를 제외하고 활용가능한 기존의 자료
  - 사례기록물, 상담일지, 정부자료, 간행물, 발표논문, 패널자료 등
  - 시간과 비용을 절감할 수 있으나, 정확도/신뢰도/타당도가 낮을 수 있음

# 1차 자료 수집 방법

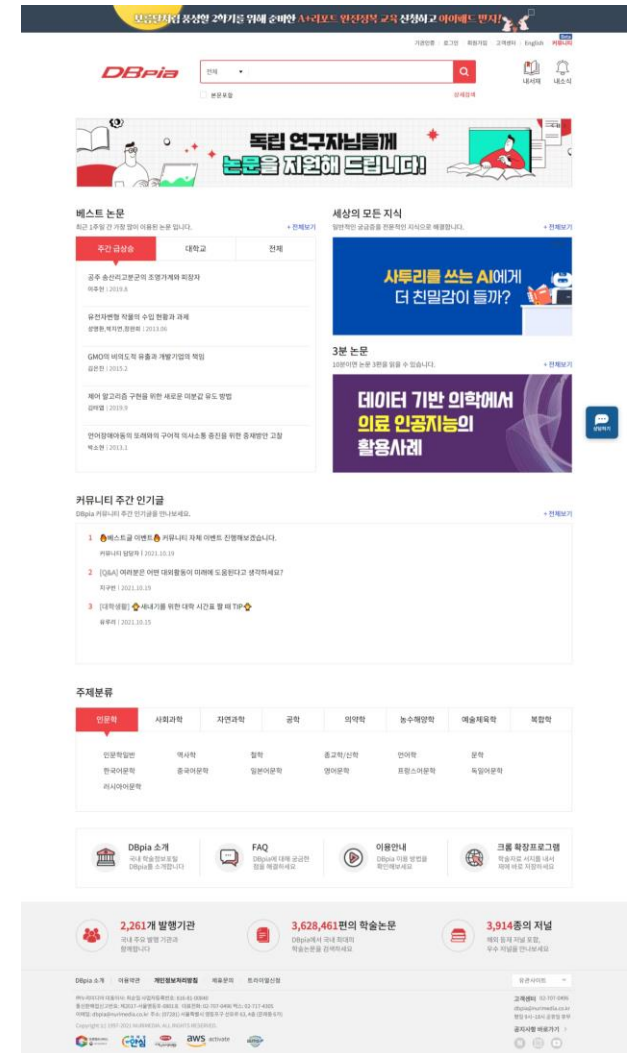
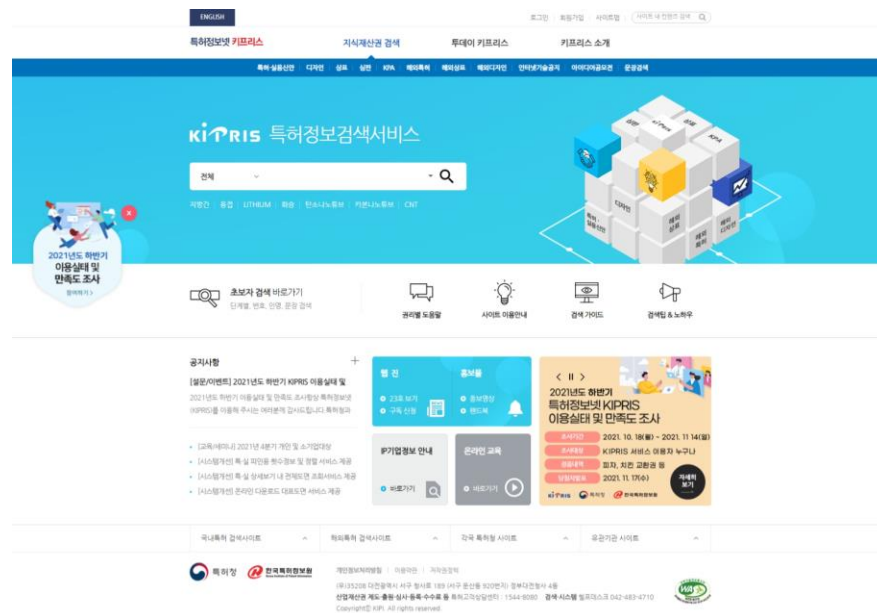
표 6.2 대표적인 자료 수집 방법

구분	특징	장점	단점
질문지법	개인적 태도와 의식조사에 주로 사용	시간과 비용이 절약되고 자료분석의 기준이 명확함	응답자의 회수율이 저조함
참여관찰법	의사소통이 어려운 경우 현지 조사에 많이 사용	언어로 표현하기 어려운 것도 조사가 가능	관찰자의 편견 개입의 우려
실험연구법	변수의 개입을 통해 다른 변수에 대한 영향을 관찰	인과 관계의 확실성이 높아짐	실험실에서만 이루어져 현실성이 우려됨
면접법	대상을 직접 만나 질의·응답이 가능	보다 깊이 있는 조사가 가능	면접자의 주관 개입될 가능성이 있음

# 2차 자료 수집 방법 - 논문 및 특허 검색



- 국내 논문
  - [www.dbpia.co.kr](http://www.dbpia.co.kr) (논문지)
  - [www.riss.kr](http://www.riss.kr) (국내 학위논문)
- 해외 논문
  - [scholar.google.com](http://scholar.google.com)
- 특허
  - [www.kipris.or.kr](http://www.kipris.or.kr) (국내특허)
  - [patents.google.com](http://patents.google.com) (국제특허)



# 역공학(Reverse Engineering, 리버스 엔지니어링)

- 기존의 완제품을 분해하거나 분석하여 설계 내용을 추적하는 방법
  - 기존의 상품, 조립체 등을 관련 도면 없이 복사해내는 과정
  - 완제품을 하나하나 뜯어 보면서, 작동되는 중간 과정을 추정/입증, 전체를 유추해 나감
  - 가전제품, 기계, 전자, 소프트웨어 등의 분야에서 쓰이는 공학적 기법
- 역공학을 시행하는 경우
  - 제품의 성능/기능 향상을 위한 새로운 방법 모색
  - 경쟁사 제품의 장단점을 분석하고 벤치마킹
  - 원래부터 설계 관련 자료가 없었거나 중간에 손실된 경우 이를 복구
  - 낡은 공정의 향상

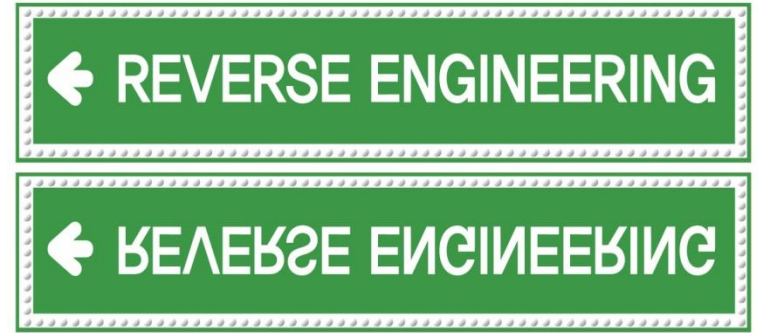
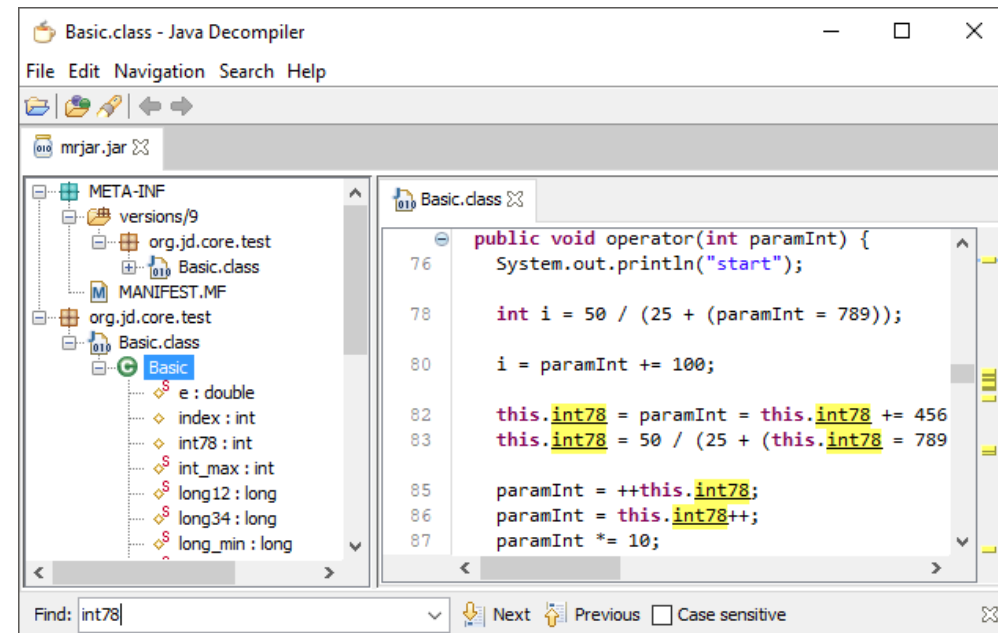


그림 6.5 역공학의 표현



# 소프트웨어 분야의 리버스 엔지니어링

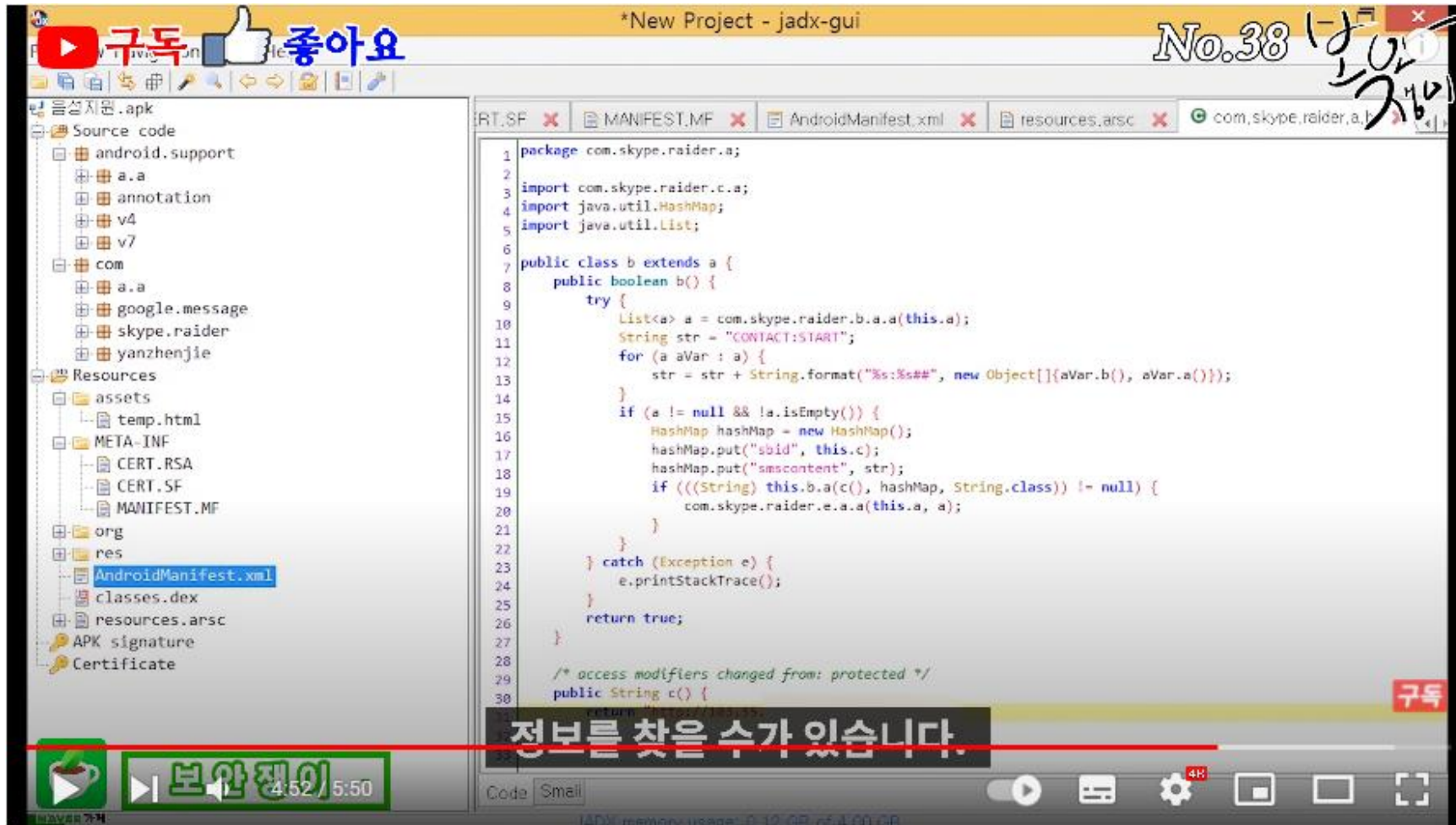
- 리버스 엔지니어링 방법
  - 디스어셈블러, 디컴파일러 등을 통해 원본 소스 복구
  - 디버깅 / 분석 / 프로파일링 툴을 통해 사용 명령어 후킹
- 리버스 엔지니어링 사례
  - 바이러스 분석을 통한 백신 개발
  - 파일 포맷의 구조 공개 (doc, alz 등)
  - 온라인 게임의 프리 서버 등



Java Decompiler ([java-decompiler.github.io](http://java-decompiler.github.io))

# 리버스 엔지니어링 사례 - APK 분석

- [몸캠 악성코드 APK 파일 분석하기 : 디컴파일 자바\(Java\) 소스코드 열어 보기 - YouTube](#)

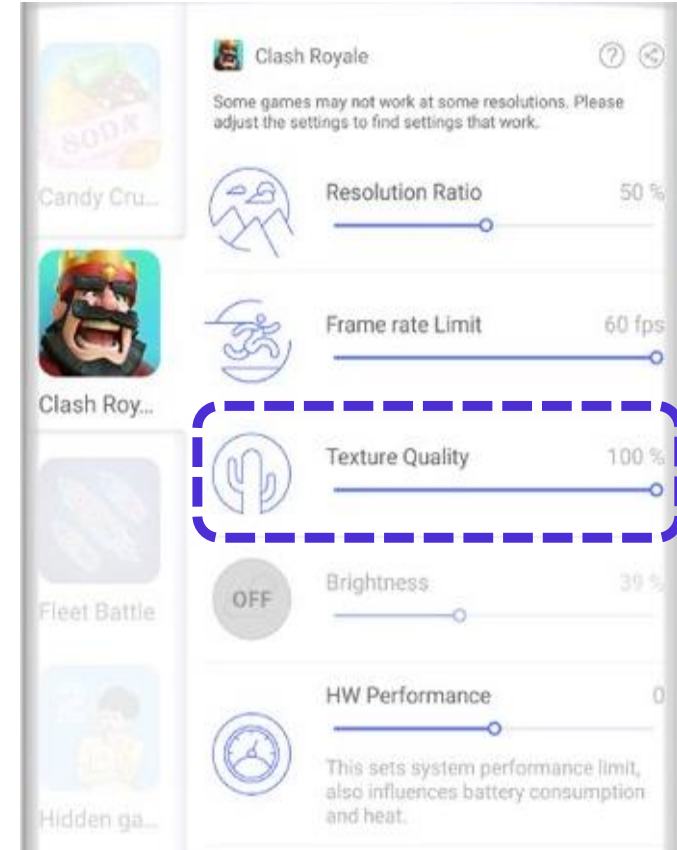




# 리버스 엔지니어링 사례 – 텍스처 크기 조절



- 미션: 삼성 게임 튜너의 기능을 분석하고, 더 좋은 것을 만들어 내거라!
- 리버스 엔지니어링을 통한 분석
  - 갤럭시 S8에서 '텍스처 퀄리티 조절' 기능을 켜 후 게임 테스트
  - Mali Graphics Analyzer를 통해, 해당 기능 사용시 GPU에 보내는 OpenGL ES 명령어를 후킹하여 모두 캡처
  - 캡처 데이터를 토대로 동작 메카니즘 파악
  - 화질 저하, 낮은 적용 범위, 앱 로딩시간 증가 등의 문제점 확인
- 더 나은 방법을 골똘히 연구
  - 텍스처 타입별로 각기 다른 방법을 적용해 보자!
  - 화질 저하를 최소화하고, 더 광범위하게 적용 가능한 기술 개발

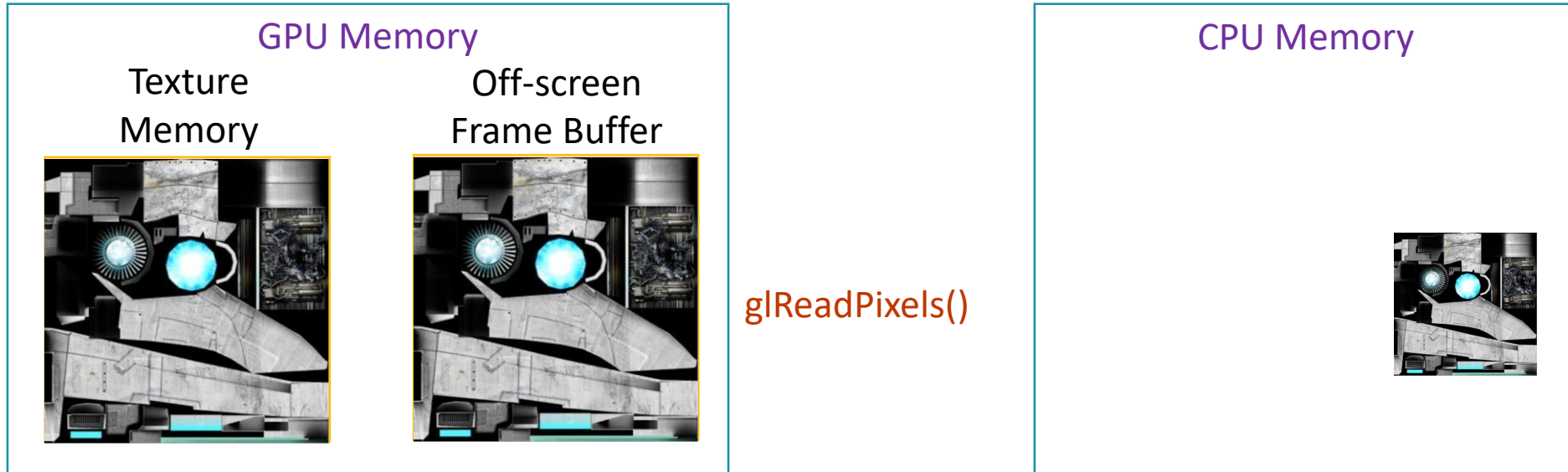


Samsung Game Tuner

[Classified Texture Resizing for Mobile Devices – SIGGRAPH 2018 \(nahjaeho.github.io\)](https://github.com/nahjaeho/Classified-Texture-Resizing-for-Mobile-Devices)

# 리버스 엔지니어링 사례 – 텍스처 크기 조절

- 역공학을 통해 분석한 게임튜너 v2.3의 텍스처 크기 조절 방식

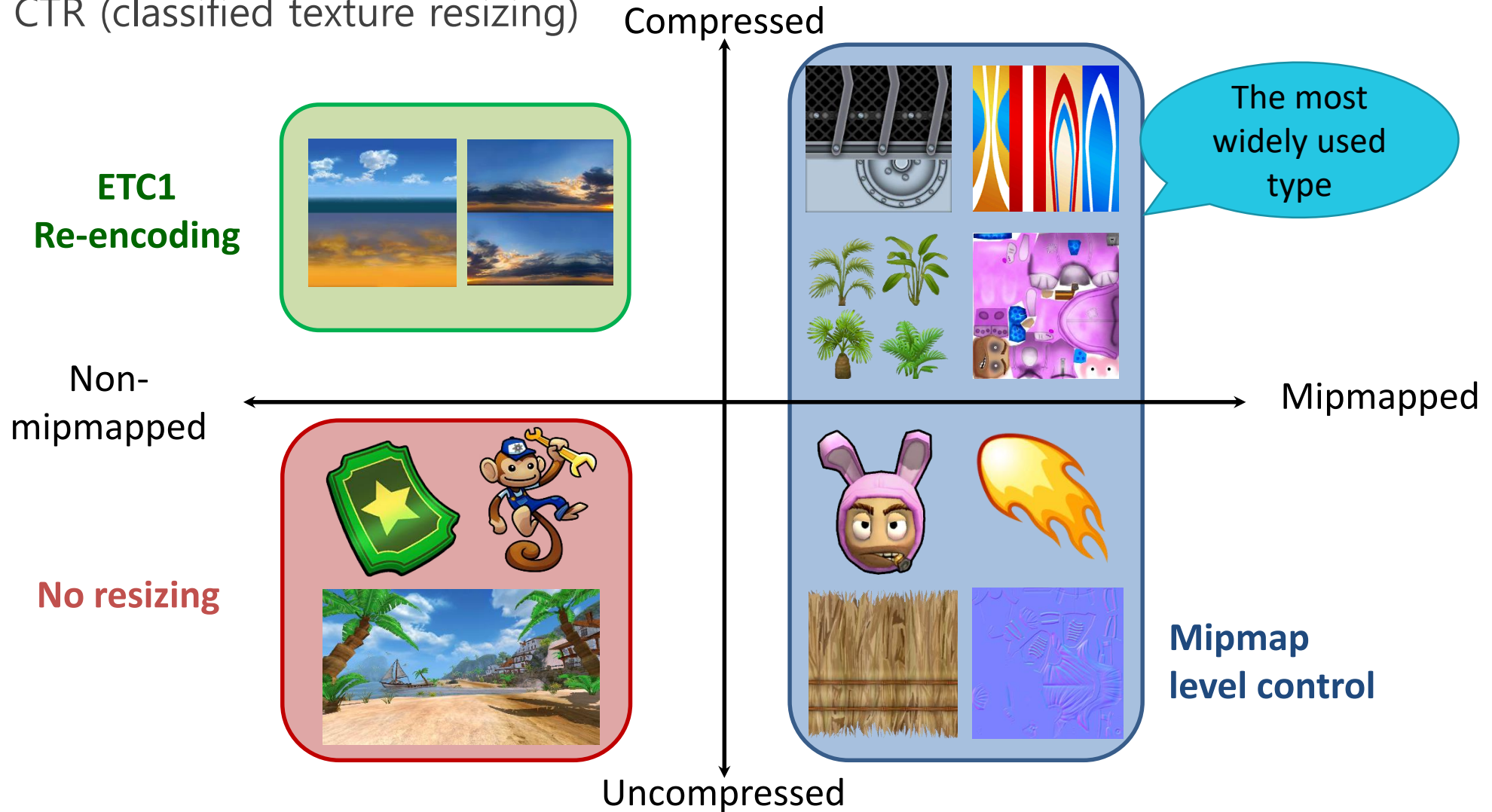




# 리버스 엔지니어링 사례 - 텍스처 크기 조절



- 제안하는 CTR (classified texture resizing)



# 리버스 엔지니어링 사례 - 텍스처 크기 조절

- 원본 이미지





# 리버스 엔지니어링 사례 - 텍스처 크기 조절

- CTR 사용시





# 리버스 엔지니어링 사례 - 텍스처 크기 조절

- 게임튜너 사용시





# 해결책 생성 등의 단계

# 해결책 생성 및 분석과 선택



- 해결책 생성
  - 새로운 아이디어, 도구, 방법들을 결합하여 독특한 해결법을 구상
  - 브레인스토밍 등의 확산적 사고와 토의 활용
- 분석과 선택
  - 주어진 해결법들을 여러가지 측면에서 분석
  - 각 분석 속성에 상대적인 비중을 적용, 가장 바람직한 방법을 최종적으로 결정
  - 예) 알루미늄 캔을 찌그러뜨리는 장치를 개발할 경우,  
기능성 30%, 편의성 25%, 안전성 20%, 내구성 10%, 시장성 15%
  - 모든 조건과 주어진 상황을 컴퓨터에 입력하여 시뮬레이션하는 방법도 사용 가능

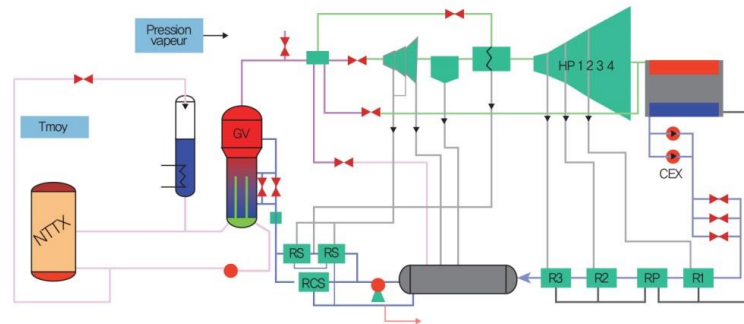


그림 6.7 원자력 발전을 위한 통합 플랫폼 시뮬레이션

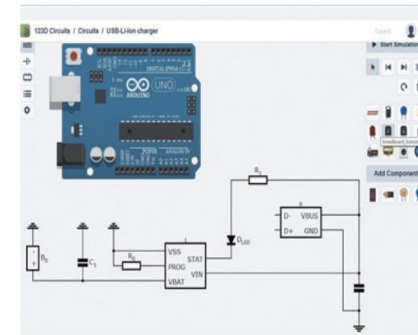
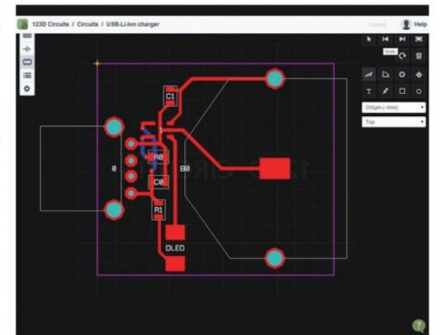


그림 6.8 회로 설계를 위한 시뮬레이션의 예





# 프로토타입(Prototype, 원형 샘플) 만들기 단계

- 구현하고자 하는 시스템과 유사한 기능을 수행할 수 있는 물리적 시스템
  - 그리스어 프로토타이폰에서 유래 ('최초', '원초적 형태'의 의미)
  - 최근에는 3D 프린터를 이용하여 빠르고 간편하게 만들 수 있음
- 개발초기에 간단한 원형을 만들어 주요 기능의 작동 여부를 테스트하고, 기능의 추가, 변경 및 삭제 등을 반영, 이 원형을 개선해 나가면서 작업 진행



그림 6.9 프로토타입 비행기들

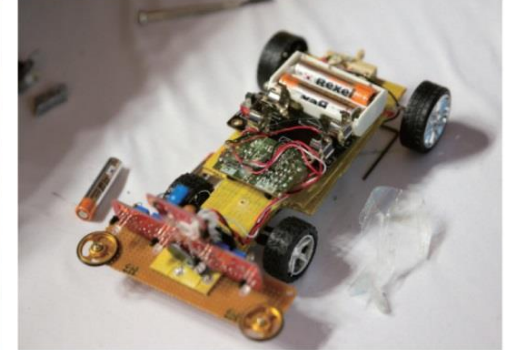


그림 6.10 축소 형태의 프로토타입 자동차

# 프로토타입의 점검 요소 및 유형

- 프로토타입에서 점검할 수 있는 요소들
  - 실제 환경에서의 여러 변수들의 성능
  - 목표 값에 맞게 작동하는지의 여부
  - 제한조건의 준수 여부
  - 구성상의 적절성 여부
  - 조립 기술과 조립 과정 점검
  - 외관, 투명도, 내구성 등 점검
- 프로토타입의 3가지 유형
  - 크기, 제한점, 변수 등이 정확히 같은 실제 모델
  - 특정한 기능을 테스트하기에 적합한 모델
  - 설계에서의 한 두가지 규칙을 고의적으로 약간 위반한 모델



# 프로토타입 만들기 4단계

[1단계] 기본적인 사용자 요구사항 분석

[2단계] 도출된 요구사항을 만족시키는 프로토타입 개발

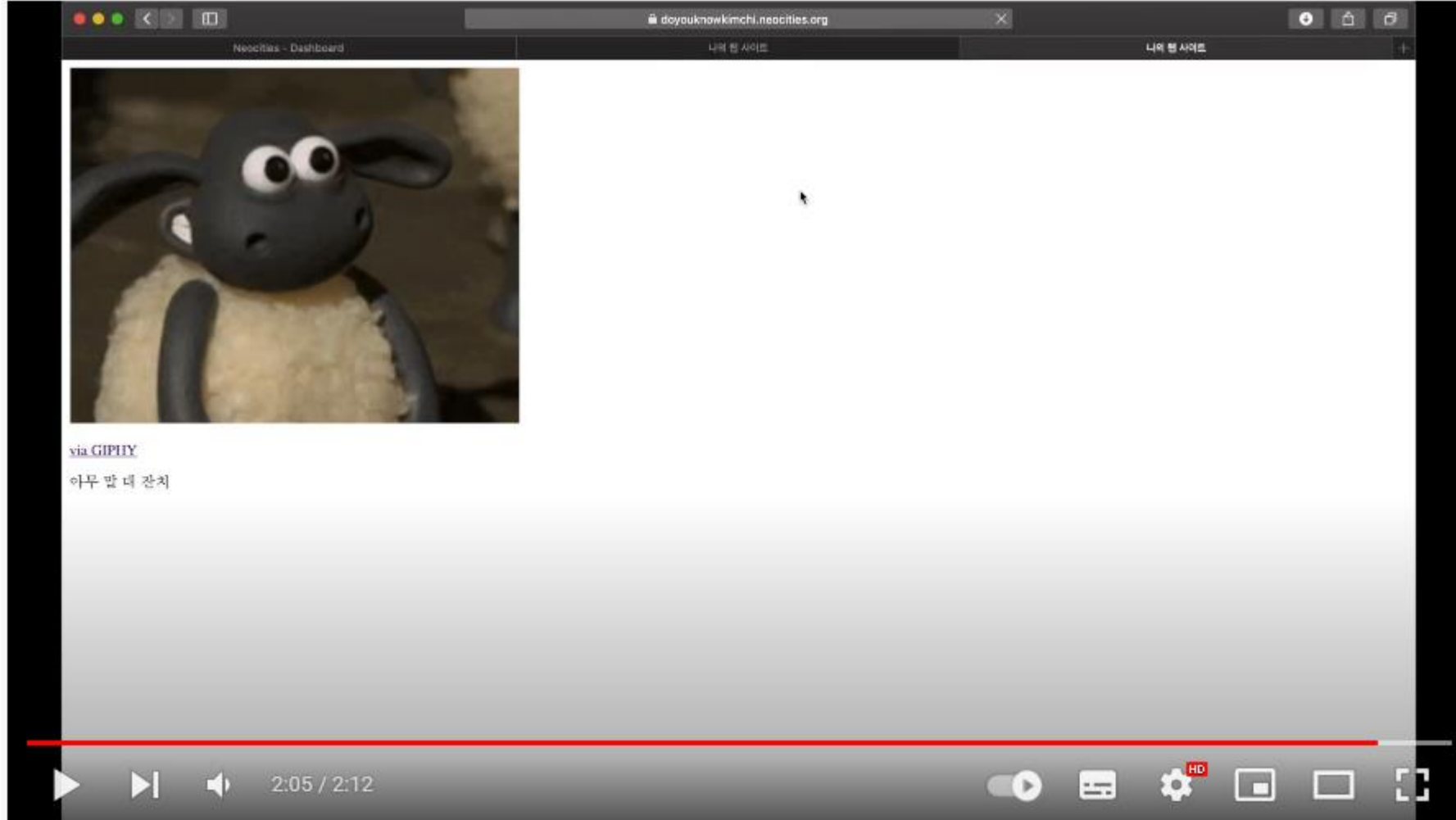
[3단계] 실제 사용해본 후 요구사항 이행 여부 확인, 보완점 제안

[4단계] 수정과 보완 시행, 사용자가 만족할 때까지 3~4단계 반복



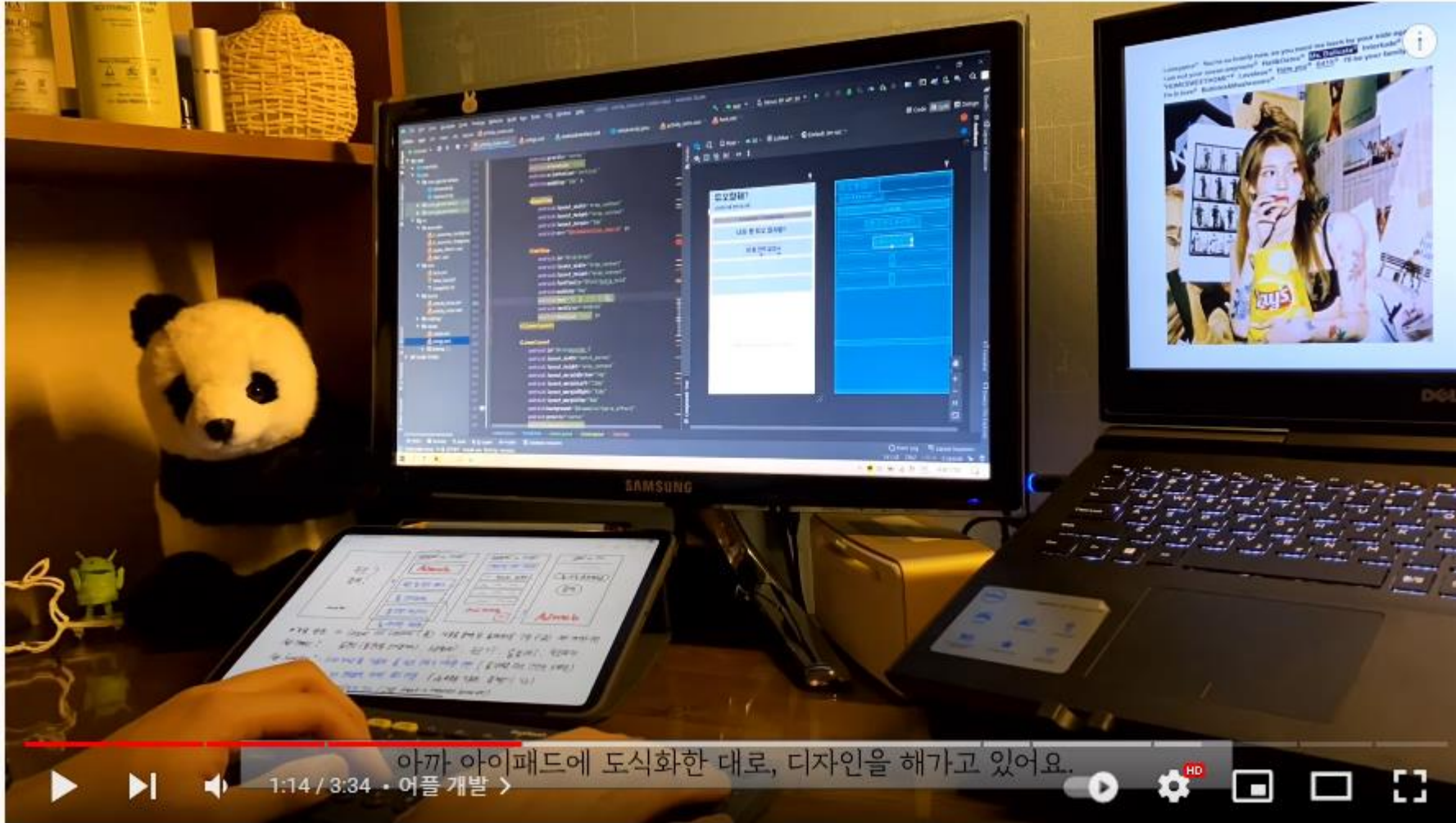
# 프로토타입 만들기 - 웹사이트 만들기

- [2분 만에 웹사이트 만들기 - YouTube](#)



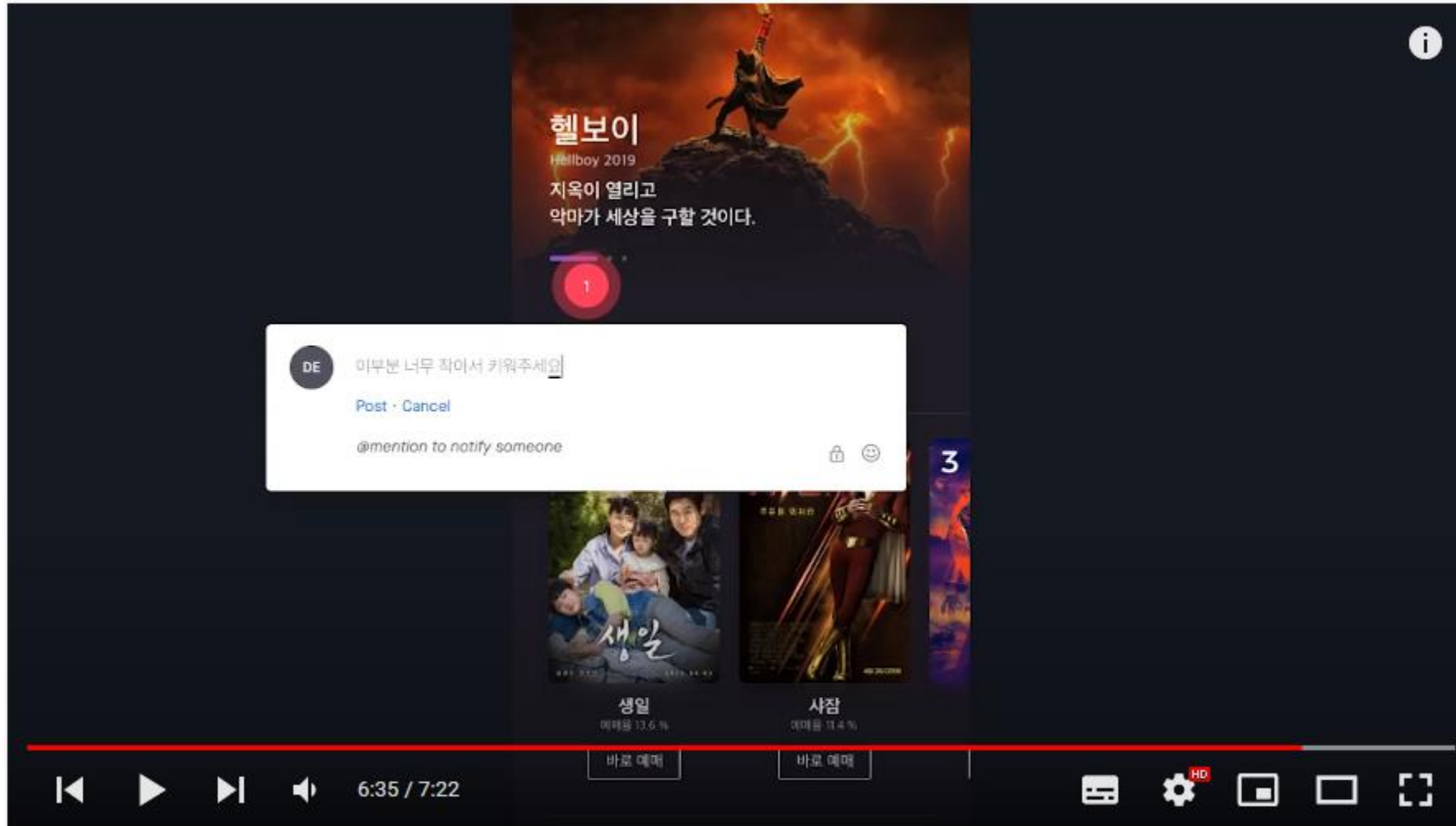
# 프로토타입 만들기 – 앱 만들기

- 취미로 앱개발하는 공돌이의 "4시간만에 앱만들기 V-Log" - YouTube



# 프로토타입 만들기 – 앱 UI 프로토타이핑

- [#1 인비전의 장점과 사용법 5분으로 끝내자 - 인비전 프로토타이핑 툴 강좌 – YouTube](#)



# 테스트와 성능 개선 단계

- 만들어진 프로토타입을 테스트하여 다양한 항목을 체크
  - 여러 가지 기능이 제대로 작동하는지 등
  - 기대한 만큼 결과가 나오지 않으면,  
[4단계]로 돌아가 다른 아이디어로 재설계
- 여러번의 피드백을 통해, 목표에 도달하는 수준까지 성능 개선
- SW QA (quality assurance)
  - 소프트웨어 테스트뿐 아니라, 품질 보증을 위한 전반적인 업무 담당



그림 6.12 다양한 항목의 체크리스트



# 소프트웨어 QA의 역할

- [How is the difference between LINE QA and QA? -Korean version- - YouTube](#)



# 설계의 구현과 보고서 작성

- 설계를 구현하여 완성한 후, 이를 바탕으로 공학설계에 대한 보고서를 작성
  - 보고서는 최종 결과를 일정한 형식에 따라 항목별로 기술
  - 최종 제품에 대한 내용을 whitepaper (백서) 형태로 만들어 배포하기도 함



〈프로젝트명〉 :		담당교수 :		
소속	학과(학부)	팀구성원	성명	학번
연구 주제와 개요				
〈프로젝트 추진 결과〉				
연구진행 결과				
최종 성과				
개인별 참여도 (개인별 작성)				

공학설계입문 팀 프로젝트  
최종보고서 양식



## NVIDIA AMPERE GA102 GPU ARCHITECTURE

Second-Generation RTX

Updated with NVIDIA RTX A6000 and NVIDIA A40 Information V2.0

[nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf](https://nvidia-ampere-ga-102-gpu-architecture-whitepaper-v2.pdf)

# 동시공학 (Concurrent Engineering)

- 제품 관련 공정을 동시 병렬적으로 설계하는 방법
  - 품질, 스케줄 등의 모든 요소를, 관련 개발자들이 처음부터 통합적으로 함께 고려
  - 각 단계를 여러 팀이 모여 병렬로 동시에 설계하고 분석할 수 있음
- 장점
  - 설계 시간 단축 및 비용 절감 가능
- 단점
  - 초기 셋업 비용의 증가 가능
  - 부서간 silo로 인한 커뮤니케이션 문제 발생 가능

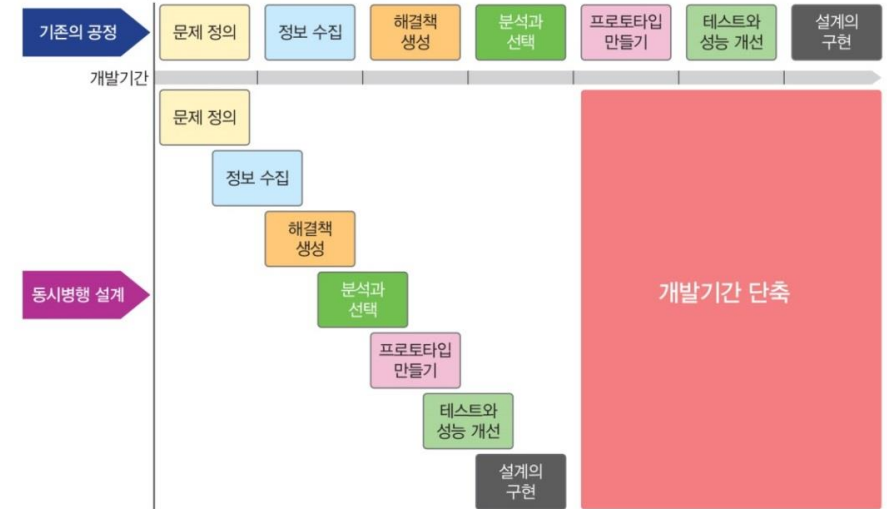


그림 6.11 동시공학의 설계 시간 단축







# 마무리

# 마무리

이번 시간에는 중간고사 풀이와 함께, 공학설계 7단계 프로세스를 심화 학습하였습니다.

- 문제 정의, 정보 수집, 해결책 생성, 분석과 선택, 프로토타입 만들기, 테스트와 성능 개선, 설계 구현

다음 시간에는 다음과 같은 내용을 다룹니다.

- 진짜 문제
- 목표 트리
- 마인드맵