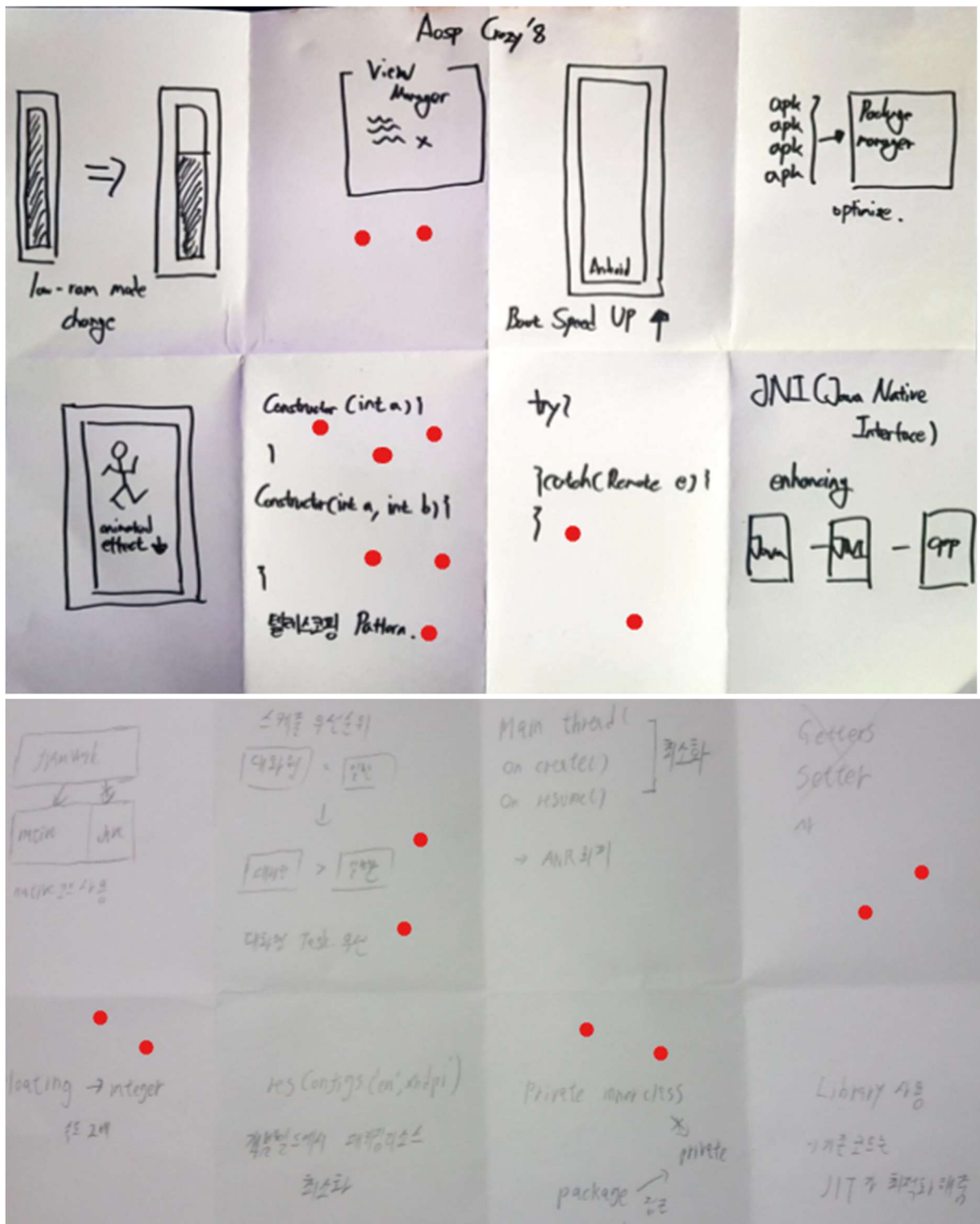


종합설계1

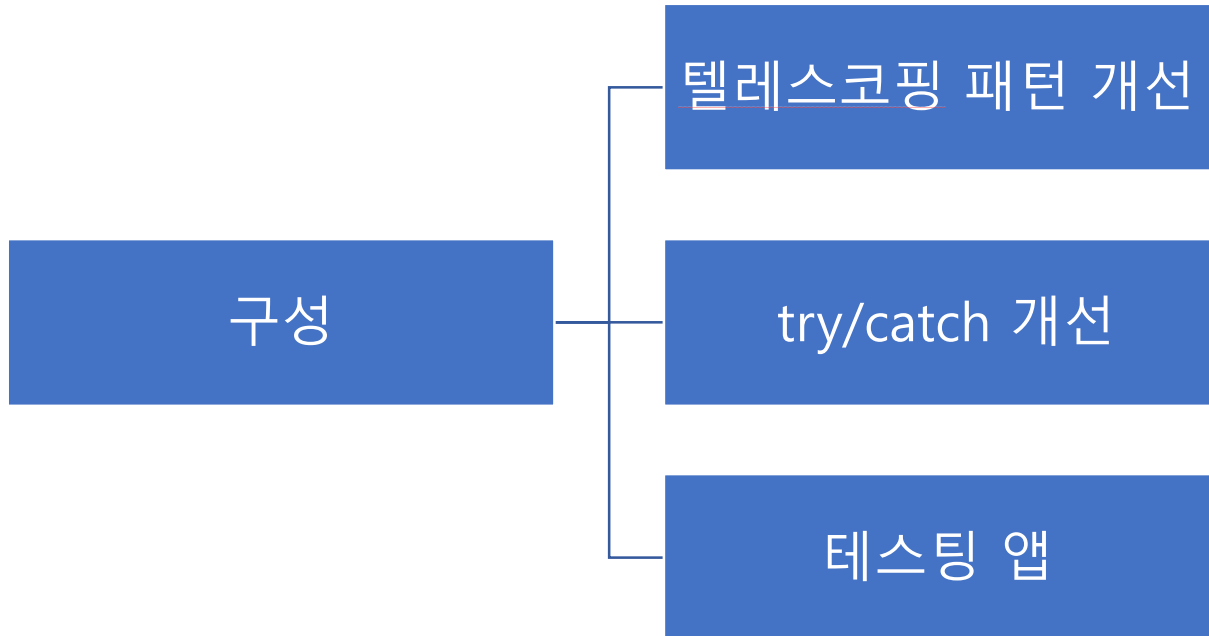
AOSP 보고서(2)

201503069

김재현

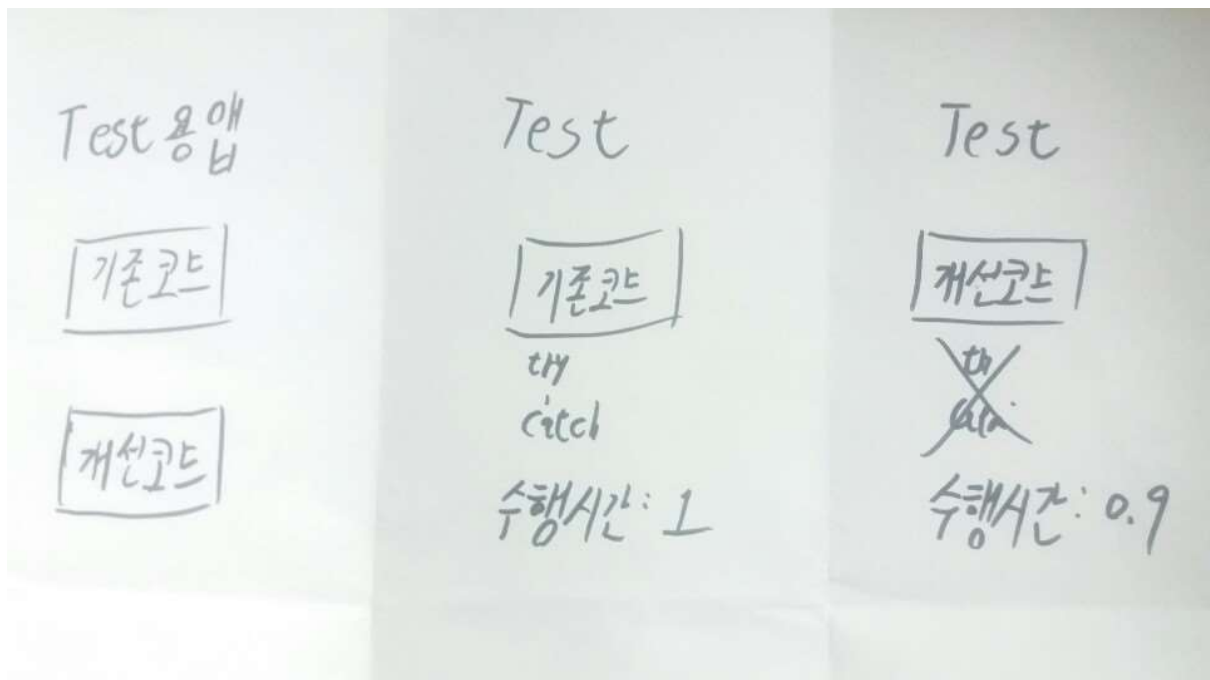


스토리보드

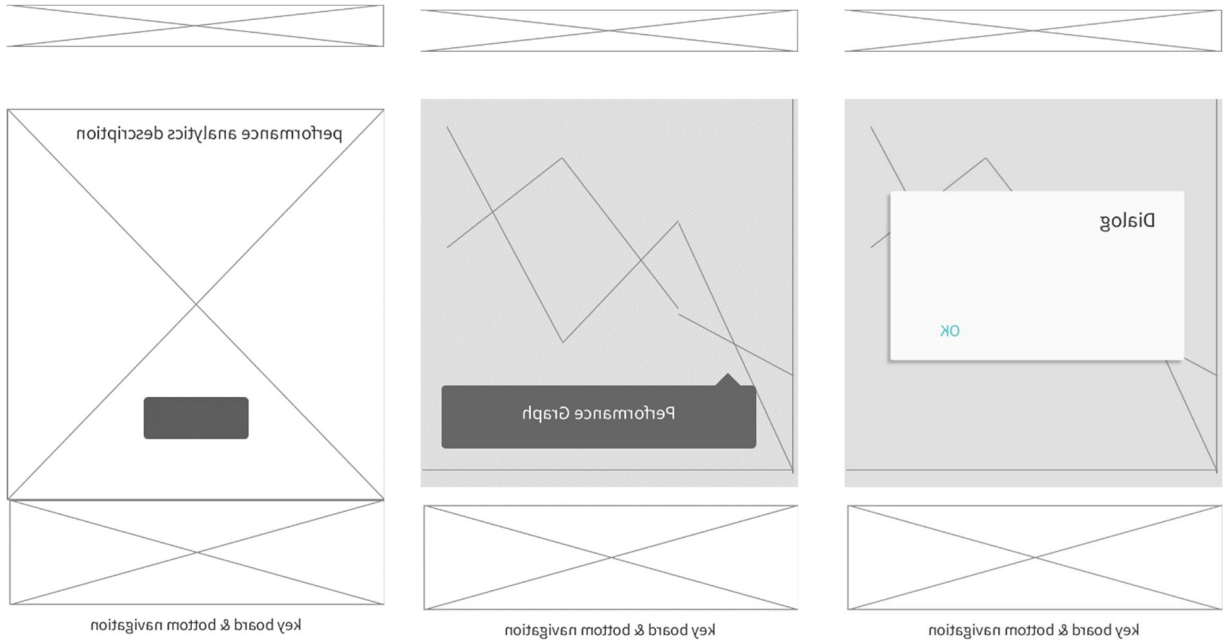


<그림 구성표>

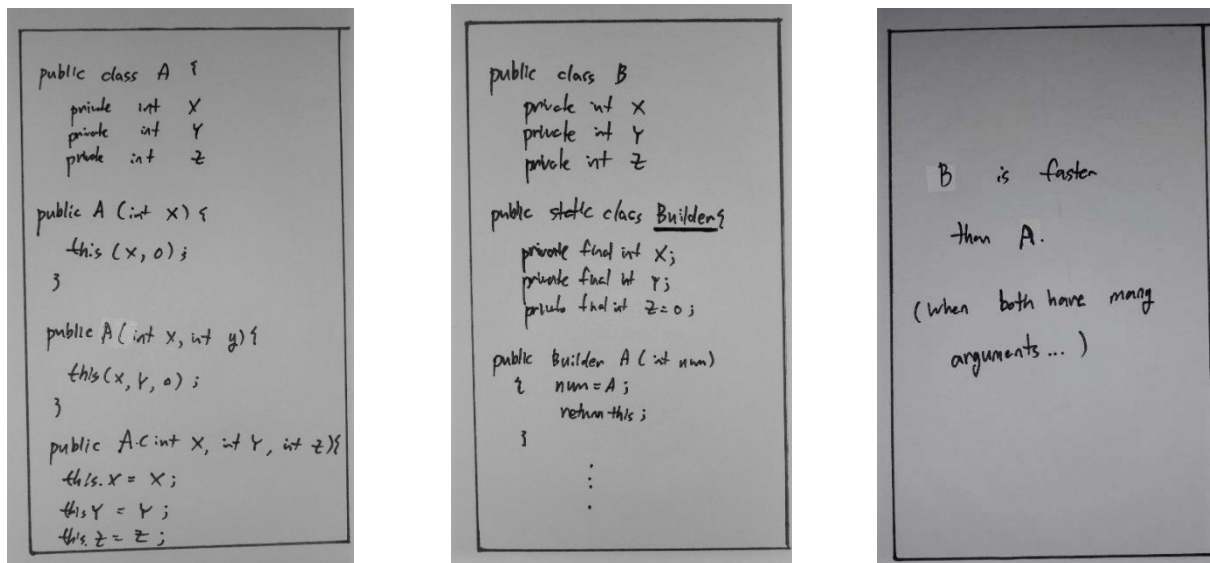
스토리보드와 프로토타입을 구성할 때 다음과 같이 3가지 기능을 통하여 구성하였다.



<그림 관습적인 예외 처리 부분 개선>



<그림 테스트 성능 측정 앱>



<그림 텔레스코핑 패턴 개선>

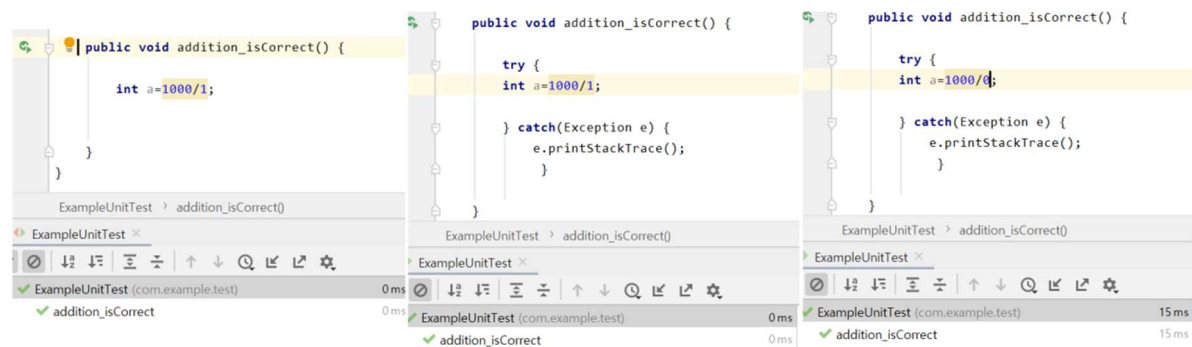
우리 AOSP 조는 안드로이드 AOSP 소스를 개선을 하기 위하여 관습적인 코드들을 개선을 하여 성능 개선을 시작을 하려 한다. 첫번째 그림은 관습적으로 예외처리를 하는 구문이지만 catch 문에서 아무런 행위를 취하고 있지 않아서 성능 적인 차이가 있는 부분을 개선을 하기 위한 것이다.

두번째는 원하는 기능들의 성능 측정을 위하여 테스트 하는 앱이다. 기존의 안드로이드 도구에서 지원을 하는 것들이 있으나 원하는 부분에서의 성능적 차이 확인하기는 어렵기 때문에 테스트 도구로써 따로 앱을 구현한다.

세번 째는 안드로이드 자바 프레임워크 단에서 텔레스코핑 패턴으로 구현된 클래스 컨스트럭터들을 빌더 패턴으로 바꾸기 위한 구성을 기획을 하였습니다. 파라미터 수가 증가를 할수록 코드 자체가 읽기 어려워지고 복잡해지기 때문에 이를 개선을 하는 것이 필요하다.

프로토타입

이를 기반으로 프로토타입을 생성을 하면



try문에서 예외 발생시 추가연산발생

```
package new1;

public class LayoutParams {

    private final int type;
    private final int flags;
    private final int format;
    private final int w;
    private final int h;
    private final int x;
    private final int y;
    private final int temp;

    public LayoutParams(int type, int flags, int format) {
        this(type, flags, format, 0);
    }

    public LayoutParams(int type, int flags, int format, int w) {
        this(type, flags, format, w, 0);
    }

    public LayoutParams(int type, int flags, int format, int w, int h) {
        this(type, flags, format, w, h, 0);
    }

    public LayoutParams(int type, int flags, int format, int w, int h, int x) {
        this(type, flags, format, w, h, x, 0);
    }

    public LayoutParams(int type, int flags, int format, int w, int h, int x, int y) {
        this(type, flags, format, w, h, x, y, 0);
    }
}
```

```
public Builder(int type, int flags, int format) { // 필수인자 생성자
    this.type = type;
    this.flags = flags;
    this.format = format;
}

public Builder w(int num) {
    w = num;
    return this;
}

public Builder h(int num) {
    h = num;
    return this;
}

public Builder x(int num) {
    x = num;
    return this;
}

public Builder y(int num) {
    y = num;
    return this;
}

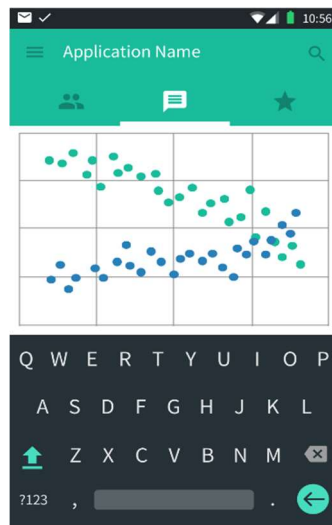
public Builder temp(int num) {
    temp = num;
    return this;
}

public LayoutParams_2 build() {
    return new LayoutParams_2(this);
}
```



성능 측정이 오래걸리 수 있으니
잠시만 기다려주세요!

측정



만들어진 스토리 보드를 기반으로 예외처리, 텔레스코핑 패턴 → 빌더 패턴으로 변경 → 테스트
앱 구현 등을 통하여 안드로이드 AOSP(Android Open Source Project) 에서의 코드 패턴 들을 개
선함으로써 성능 개선을 기획