

ぶりを使わない場合の仕様変更が発生した場合の差分

A: 「A フラグを B から C に、でも D が E だったら F に」

```
public void doUpdate(TestDto dto) {
    if (dto.getAFlag().equals("B")) {
        if (dto.getDFlag().equals("E")) {
            dto.setAFlag("F");
        } else {
            dto.setAFlag("C");
        }
    }
    TestDao dao = (TestDao) this.container.getComponent(TestDao.class);
    long id = dto.getTestId();
    if (dao.selectCountById(id) == 0) {
        dao.insert(dto);
    } else {
        dao.update(dto);
    }
}
```

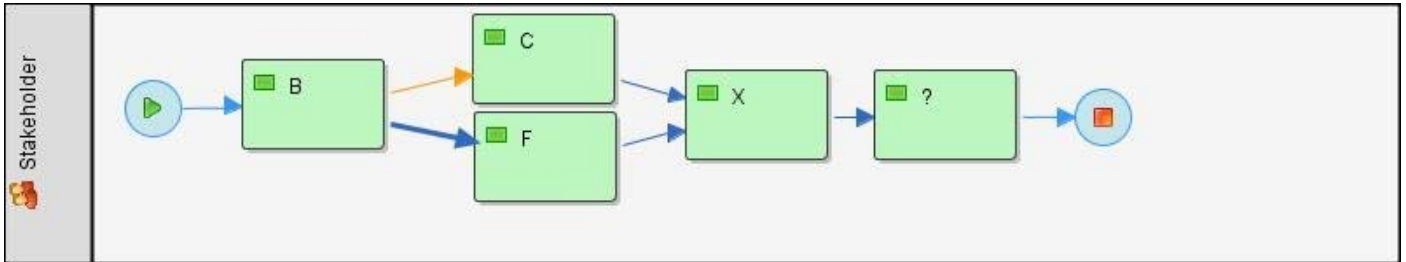
B: 「でも、D フラグが G だったら H に」

```
public void doUpdate(TestDto dto) {
    if (dto.getAFlag().equals("B")) {
        if (dto.getDFlag().equals("E")) {
            dto.setAFlag("F");
        } else if (dto.getDFlag().equals("G")) { // 追加した。
            dto.setAFlag("H"); // 追加した。
        } else {
            dto.setAFlag("C");
        }
    }
    TestDao dao = (TestDao) this.container.getComponent(TestDao.class);
    long id = dto.getTestId();
    if (dao.selectCountById(id) == 0) {
        dao.insert(dto);
    } else {
        dao.update(dto);
    }
}
```

ぶりを使った場合の仕様変更が発生した場合の差分

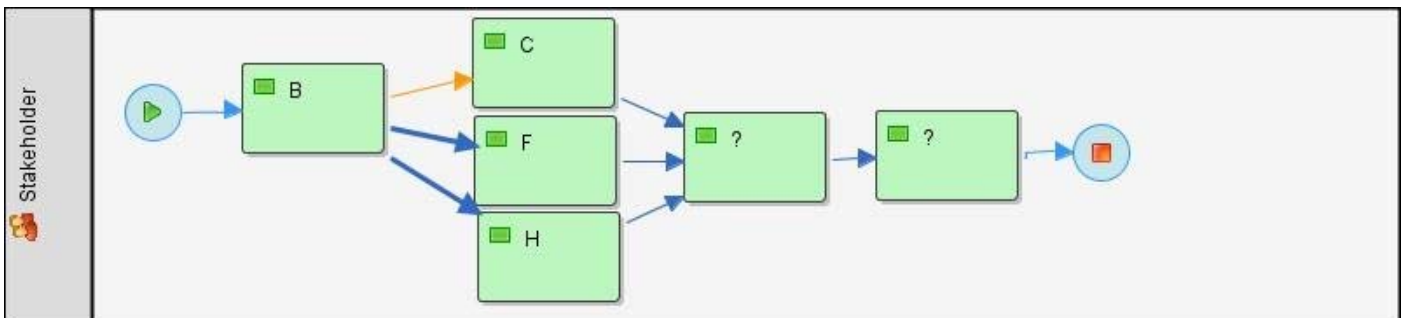
A: 「A フラグを B から C に、でも D が E だったら F に」

```
public void doUpdate(TestDto dto) {
    SimpleBuriProcessor processor = (SimpleBuriProcessor) this.container
        .getComponent(SimpleBuriProcessor.class);
    processor.toNextStatus("Package.ProcessA.B", dto);
}
```



B: 「でも、D フラグが G だったら H に」

```
public void doUpdate(TestDto dto) {
    SimpleBuriProcessor processor = (SimpleBuriProcessor) this.container
        .getComponent(SimpleBuriProcessor.class);
    processor.toNextStatus("Package.ProcessB.B", dto);
}
```



以上を踏まえて分かるコト

- ぶりを使うとコードから
 - if 文が消えました！
 - 変更の際の編集が消えました！