



# Test-Driven Development with Java

# Code แบบไหนที่ชอบ ?

# Code แบบไหนที่ไม่ชอบ ?



# PROBLEM

## Circular Buffer

[http://en.wikipedia.org/wiki/Circular\\_buffer](http://en.wikipedia.org/wiki/Circular_buffer)

# PROBLEM

## Circular Buffer

[http://en.wikipedia.org/wiki/Circular\\_buffer](http://en.wikipedia.org/wiki/Circular_buffer)

# 5 นาที

## ทำความเข้าใจกับปัญหา

# สลับคู่ทุกๆ 5 นาที

## แก้ไข code ต่อไป



# Retrospective 5 นาที

# 5 นาที

## เขียน code ต่อไป

# Retrospective 5 นาที

# Is this how Software Developers view their Customer?

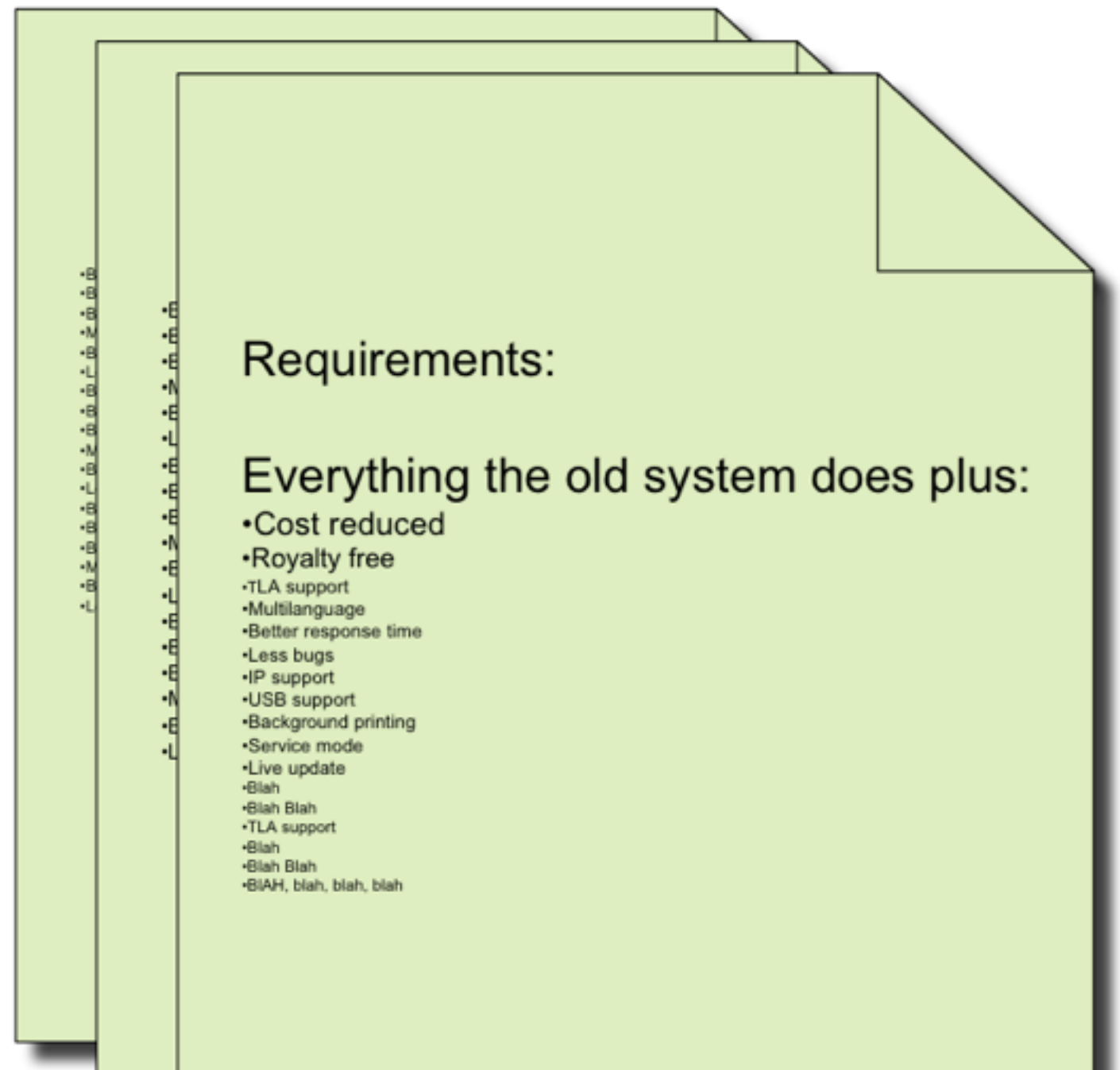
Can you add  
this new  
feature?



Go ahead and  
break other  
features while  
you are at it, no  
problem

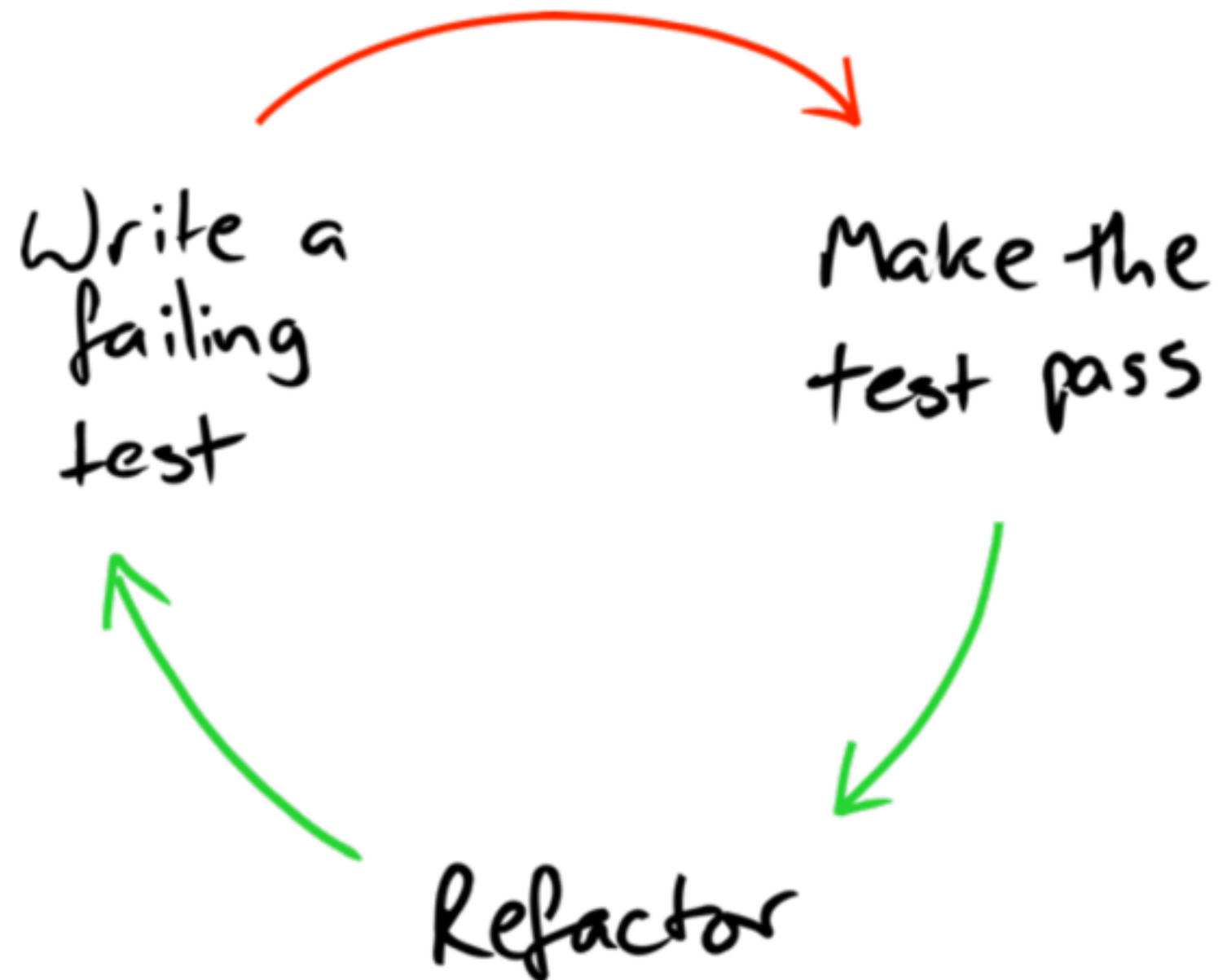
# Requirements Change

- A sign of a successful system
- Customers want new features
- Marketing wants backwards compatibility
- Old behavior must be preserved

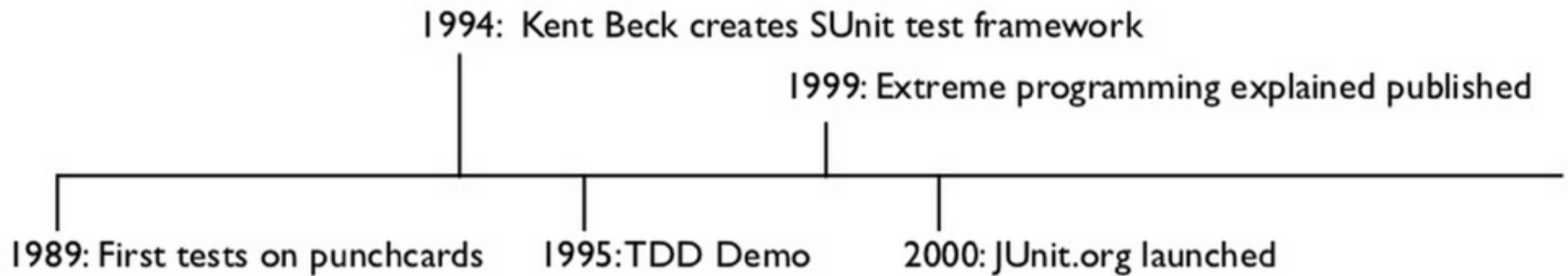


# เราต้องการวิธีการใหม่ ?

# TEST DRIVEN DEVELOPMENT



# ประวัติของ TDD





# Proactive vs Reactive

หาปัญหาก่อน

ทดสอบทีหลัง

ตรวจสอบ req ก่อนทำ

แก้ไขตาม defect ที่แจ้ง

ตรวจสอบ code

debug code

# ทำไมต้อง TDD

Software **เปลี่ยน**อยู่เสมอ

คนเรามัก **มโน** และ ทำ**ผิดพลาด**อยู่เสมอ

# ทำไมต้อง TDD

บอกคุณว่า ผิดตรงไหน

ไม่ต้องเสียเวลา **debug**

# ทำไมต้อง TDD

Test นั้นช่วยให้คุณสร้างสิ่งที่ **นโม**  
และสร้างความคาดหวังขึ้นมา

# ทำไมต้อง TDD

ผลที่ได้คือ Testable design

Loosely coupled, High cohesion

**STEVE FREEMAN & NAT PRYCE**

**Never write new functionality  
without a failing test**

# ROBERT C. MARTIN

Testing is about **trust**

# SIMPLE DESIGN RULES

All the tests pass

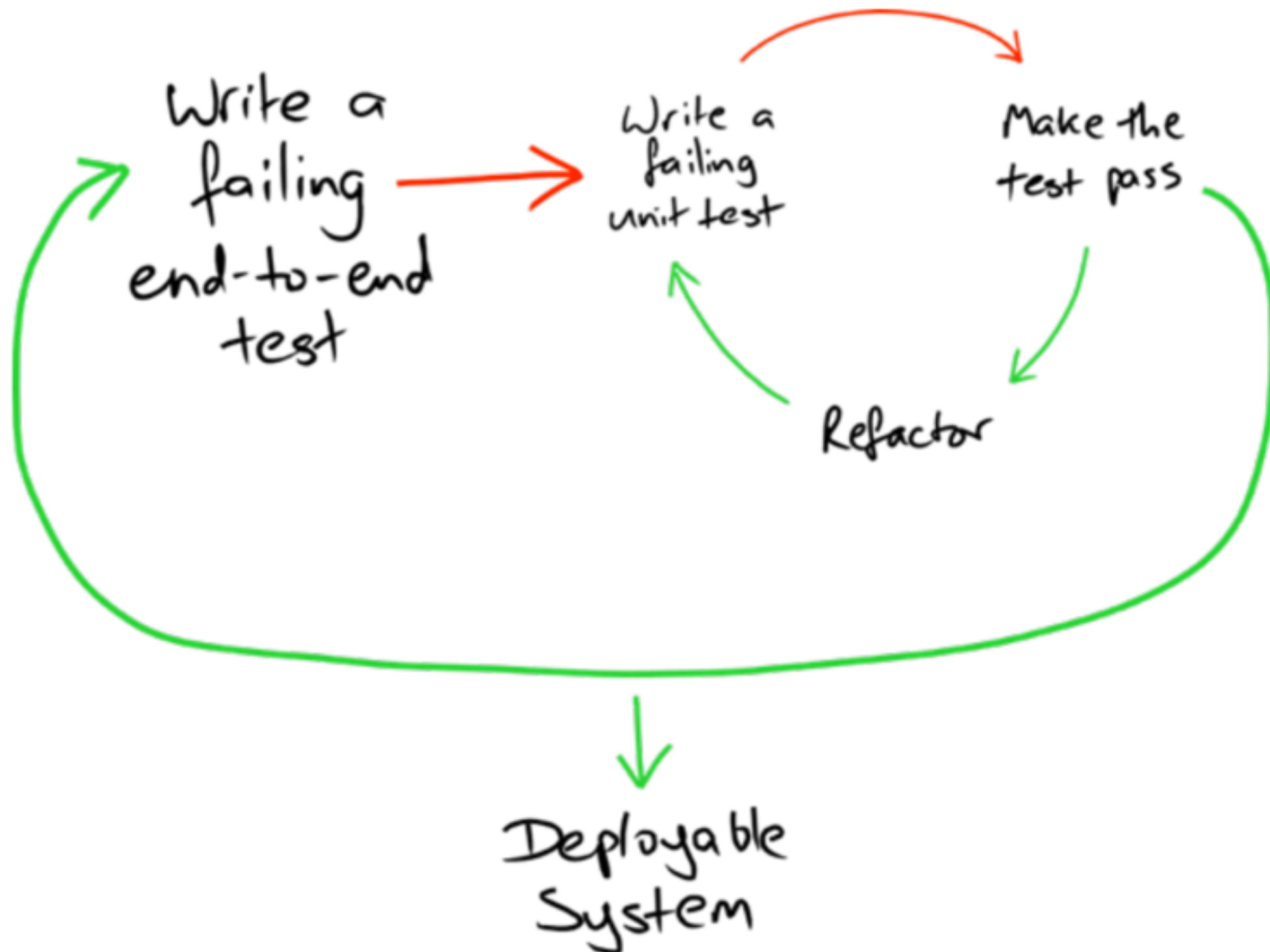
There is no duplication

Focus your intent

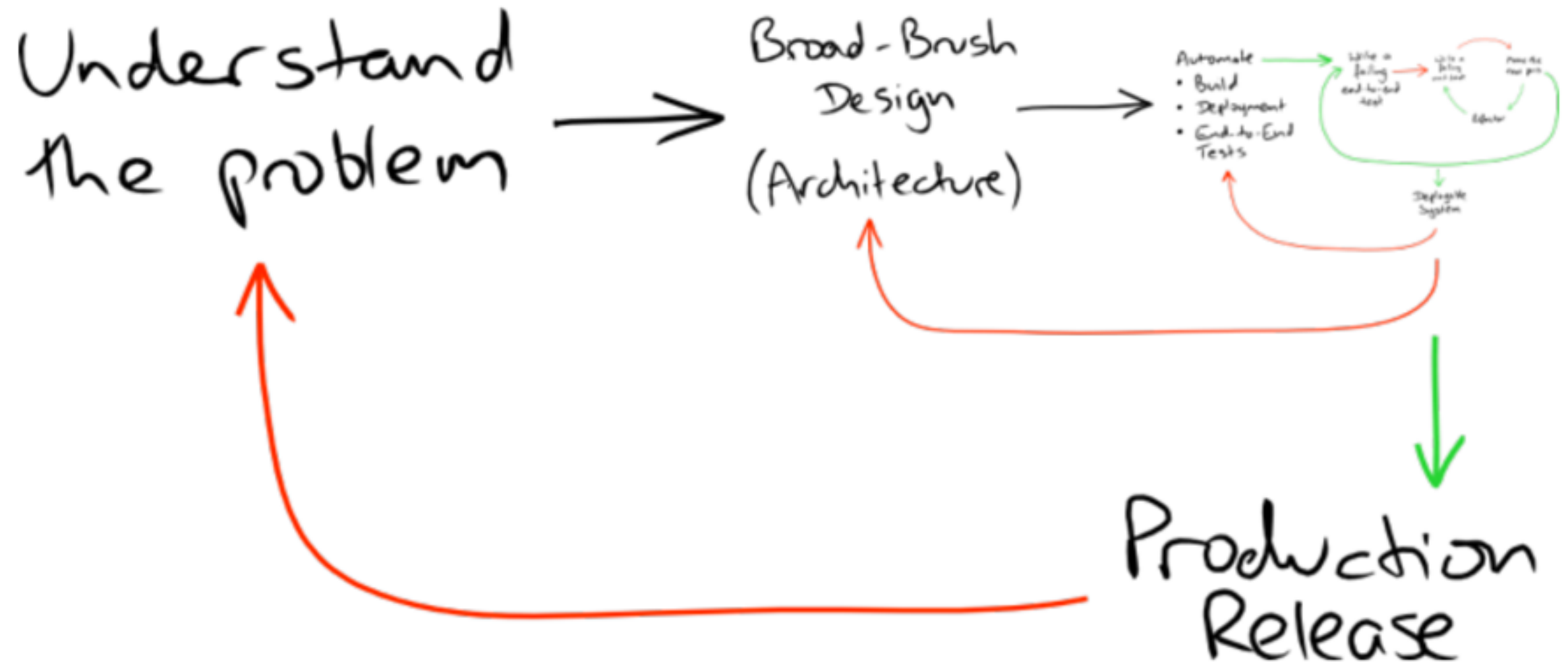
Classes and methods are minimized



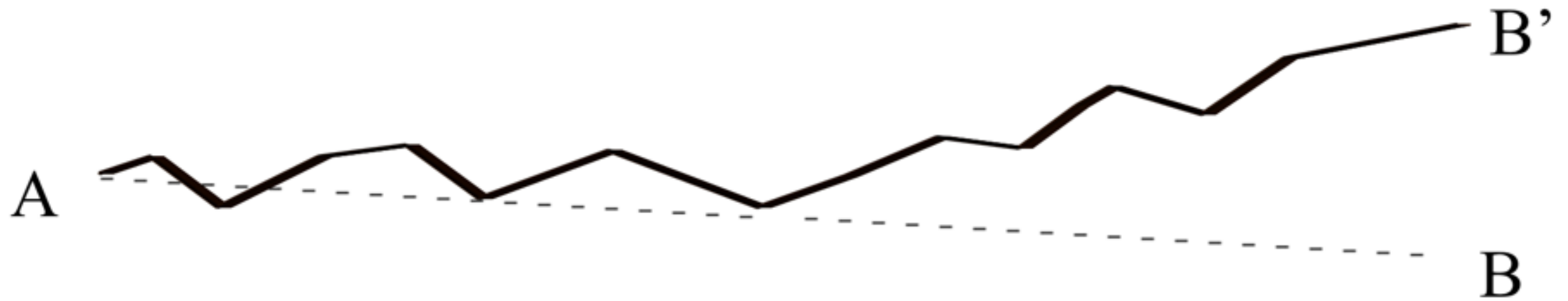
# ACCEPTANCE TEST DRIVEN DEVELOPMENT



# LARGER FEEDBACK LOOP



# Small Step



# ก่อนจะเริ่มเขียน Test

ทำการออกแบบก่อน

# 5 นาที

1. มี API หรือ method ชื่ออะไรบ้าง
2. Data structure ง่ายที่สุดที่จะใช้งาน
3. การทำงานภายในของระบบ ที่จะเกิดขึ้น

# Test แรกที่จะเขียนคือ อะไร ?

# Empty Buffer ?

สร้าง Buffer ครั้งแรกจะต้องว่าง ?

# Empty Buffer ?

ทำการเพิ่มข้อมูล



# Empty Buffer ?

ทำการเพิ่มข้อมูล และ ดึงข้อมูล ออกมาใช้

# Full Buffer ?

สร้าง Buffer ครั้งแรกจะต้องเต็ม ?

# อะไรต่อไป ?

# เพิ่มข้อมูล

เพิ่มครึ่งละหนึ่ง

เพิ่มครึ่งละสอง

เพิ่มครึ่งละสาม

# กำหนดขนาดของ Buffer

ค่า default ของ Buffer

กำหนดค่าขนาดของ Buffer ได้เอง

# อะไรต่อไป ?

# Full Buffer

เพิ่มข้อมูลจนเต็ม

เพิ่มข้อมูลจนเต็ม และ ดึงจนว่าง

# อะไรต่อไป ?



# ตรวจสอบพฤติกรรม

เพิ่มข้อมูล

เพิ่มข้อมูล

ดึงข้อมูล

เพิ่มข้อมูล

ดึงข้อมูล

ดึงข้อมูล

# Overflow ?

# Underflow ?