



Nåleputetesting virker ikke

Geir Hedemark



basefarm

Meg – cand. scient, hardwarenisse, så software
arkitekt
Basefarm – applikasjonsdrift



Dette vil gå fort – Kan virke bastant
Åpent sinn – mine feil, ikke deres
“Jeg synes det virker bra, jeg” gyldig – jeg velger situasjonen min
Firma som har tenkt å leve mer enn 3 år, og utkonkurrere de andre



Test first programming

**“Test first programming” er det beste som har
skjedd softwarebransjen siden vi sluttet å
bruke hullkort!**

Oppklare misforståelse
Unngå at tester blir kuttet
Kent Beck er gud



Test driven design

**“Hvis du skriver unit-tester,
og så refaktorerer,
så kommer arkitekturen til å bli bra”**

Som designmetodikk?

Å stikke nåler i en pute

```
public class foo {
```

```
    public void bar () {  
        4‰  
    }
```

Si 50k linjer kode og
konfigurasjon

Metoder på 200 linjer

En test for en slik
metode dekker 4‰

Å stikke nåler i en pute

```
public class foo {
```

```
    public void bar () {  
        4‰  
    }
```

“Hvis du lager en komplett testpakke på metodenivå, for alle metoder, så faller arkitekturen din på plass slik at du får et løst koblet system”

4 promille gir alt det der? Kan jeg få 10%?

Utsagnet er en spissformulering – lov å ikke kjenne seg igjen

Reductio ad absurdum – Sokrates – “innført nye guder” “leder ungdommen på villspor”

Å stikke nåler i en pute

```
public class foo {
```

```
    public void bar () {  
        4‰  
    }
```

Jeg bomma, den røde boksen skal refaktoreres ut av metoden og inn i en annen del av systemet.

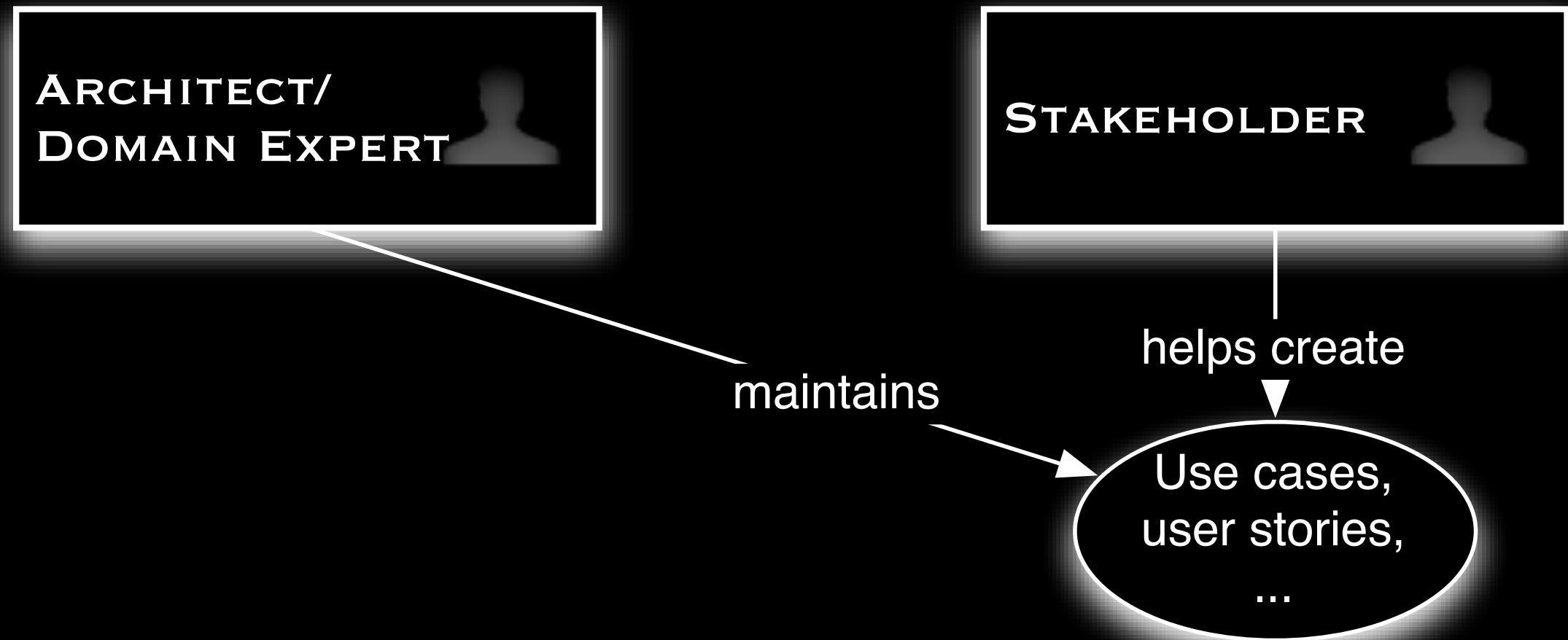
Refaktoring avhengig av tester som ikke knekker
Testen knekker! Alt som går utenfor 4 promille er off limits!
Test av implementasjon eller funksjon? “Hardt koblede tester”?
Det var nok ikke dette Kent Beck mente – smart mann



Refaktorering avhengig av tester som ikke knekker
Testen knekker! Alt som går utenfor 4 promille er off limits!
Test av implementasjon eller funksjon? “Hardt koblede tester”?
Det var nok ikke dette Kent Beck mente – smart mann



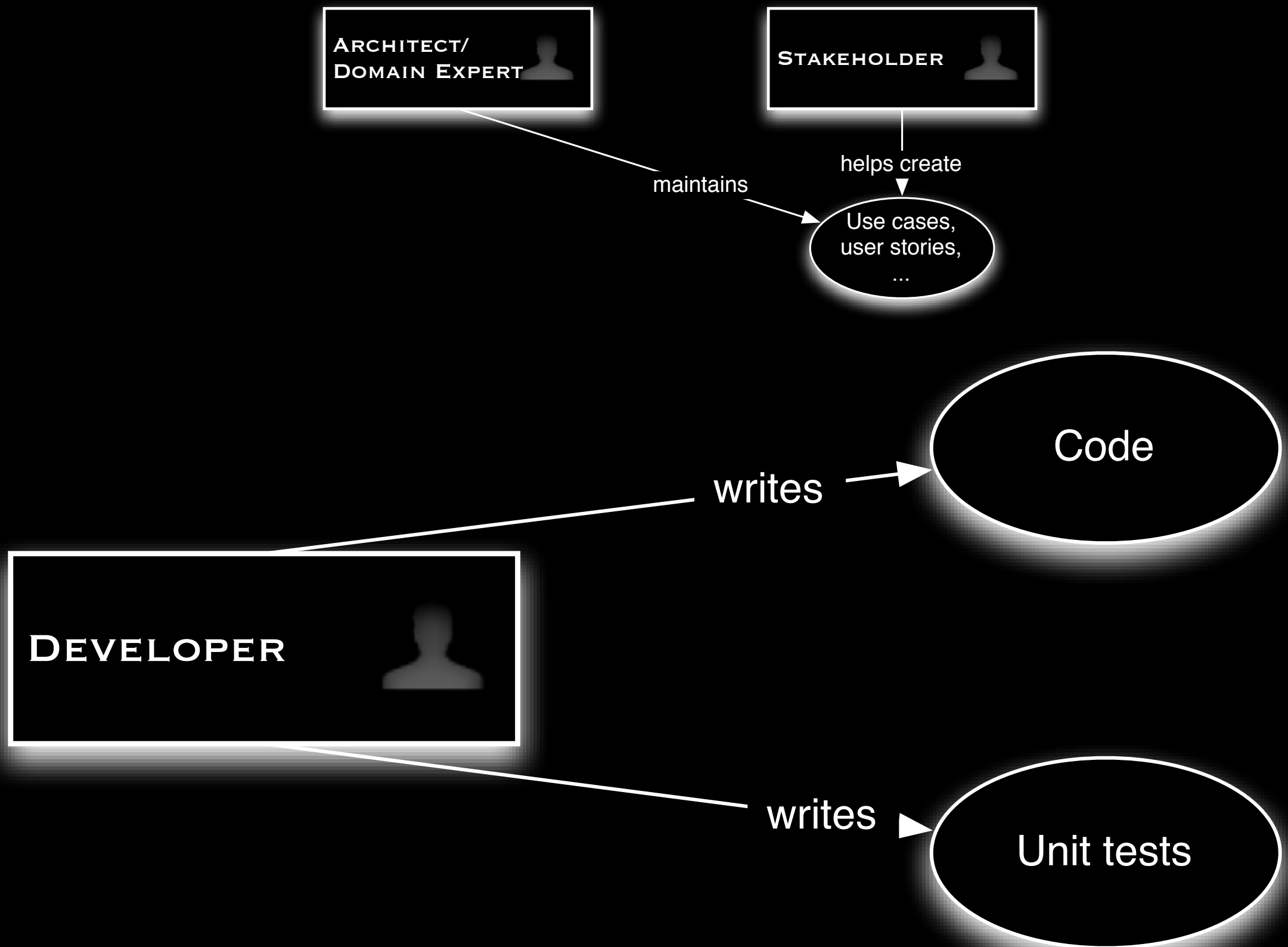
Arbeidsflyt



Overordnet flyt for fangst av funksjonalitet
Hadde det ikke vært en ide om disse menneskene snakket sammen?



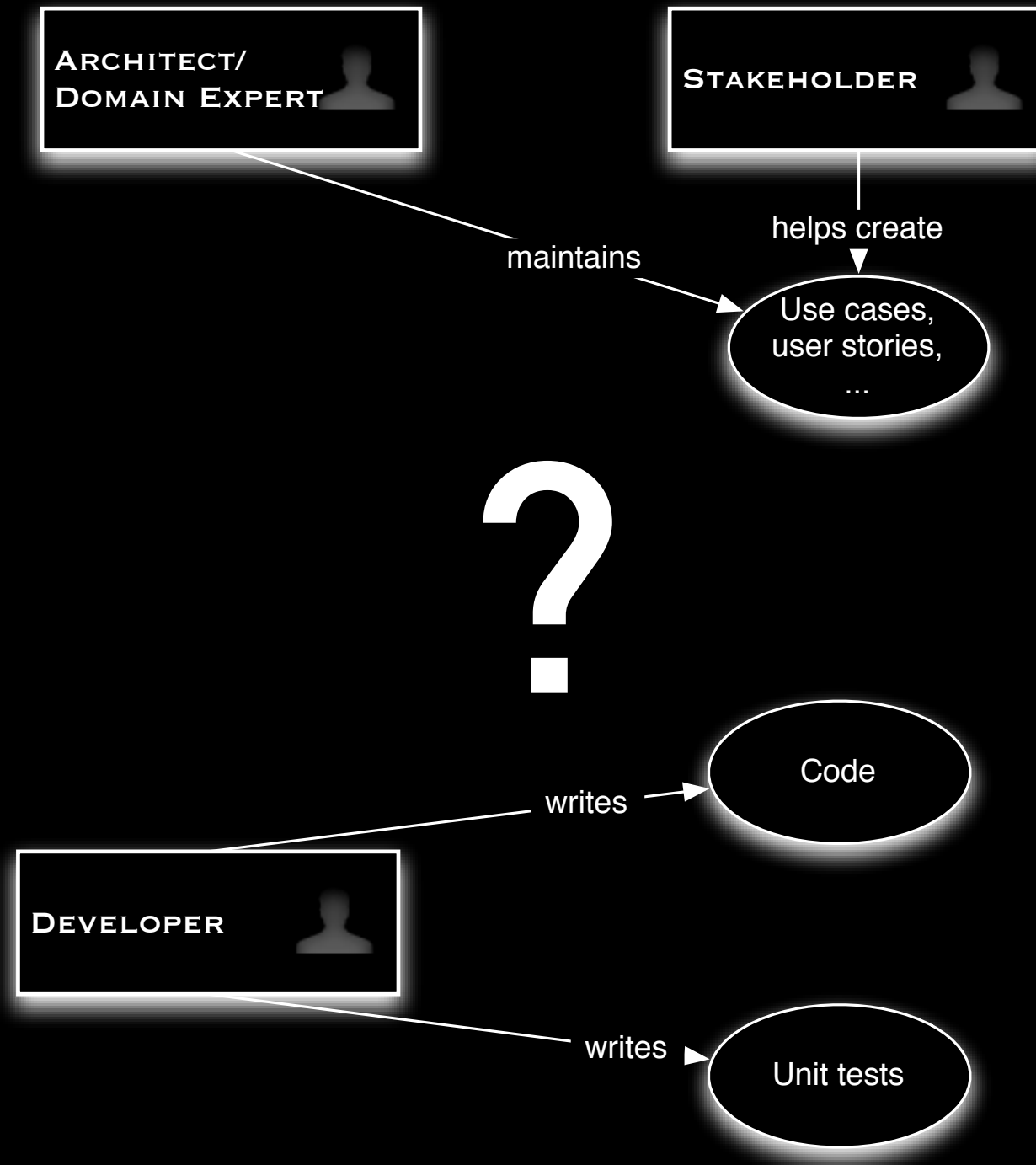
Arbeidsflyt



Overordnet flyt for fangst av funksjonalitet
Hadde det ikke vært en ide om disse menneskene snakket sammen?



Arbeidsflyt



Overordnet flyt for fangst av funksjonalitet
Hadde det ikke vært en ide om disse menneskene snakket sammen?



Mitt forslag til første steg mot løsning

Test det du får betalt for

“purring skal sendes om kunden ikke har
betalt innen 2 uker”

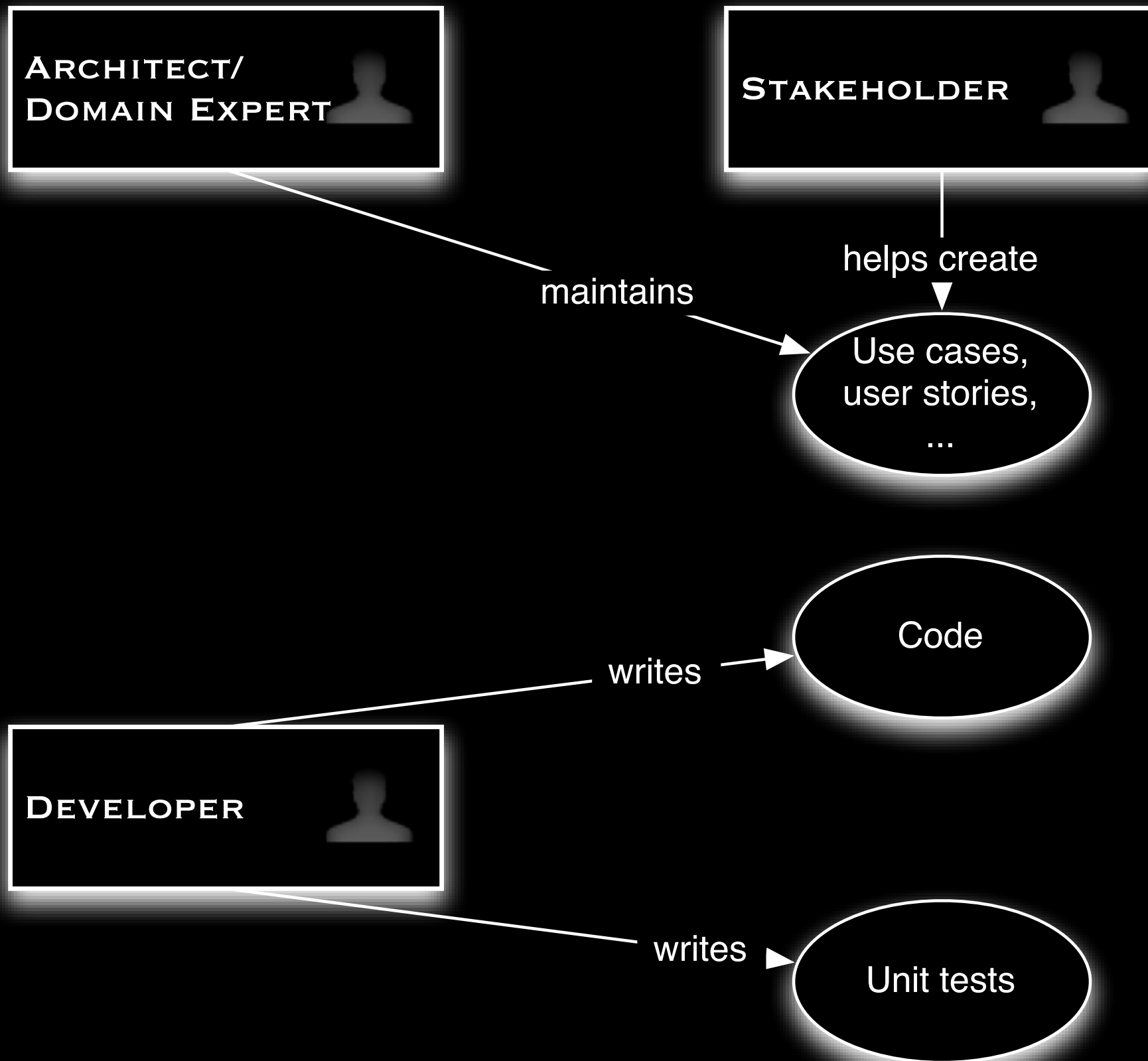
“testEmptyCustomerShouldFail” i stedet
for “testNull”

BDD, Dan North

Finkornede unit-tester – ikke det vi vil måle
Funksjonspunkter, ikke metoder!
Veldig fint å kunne linke til use case – vedlikehold use case!

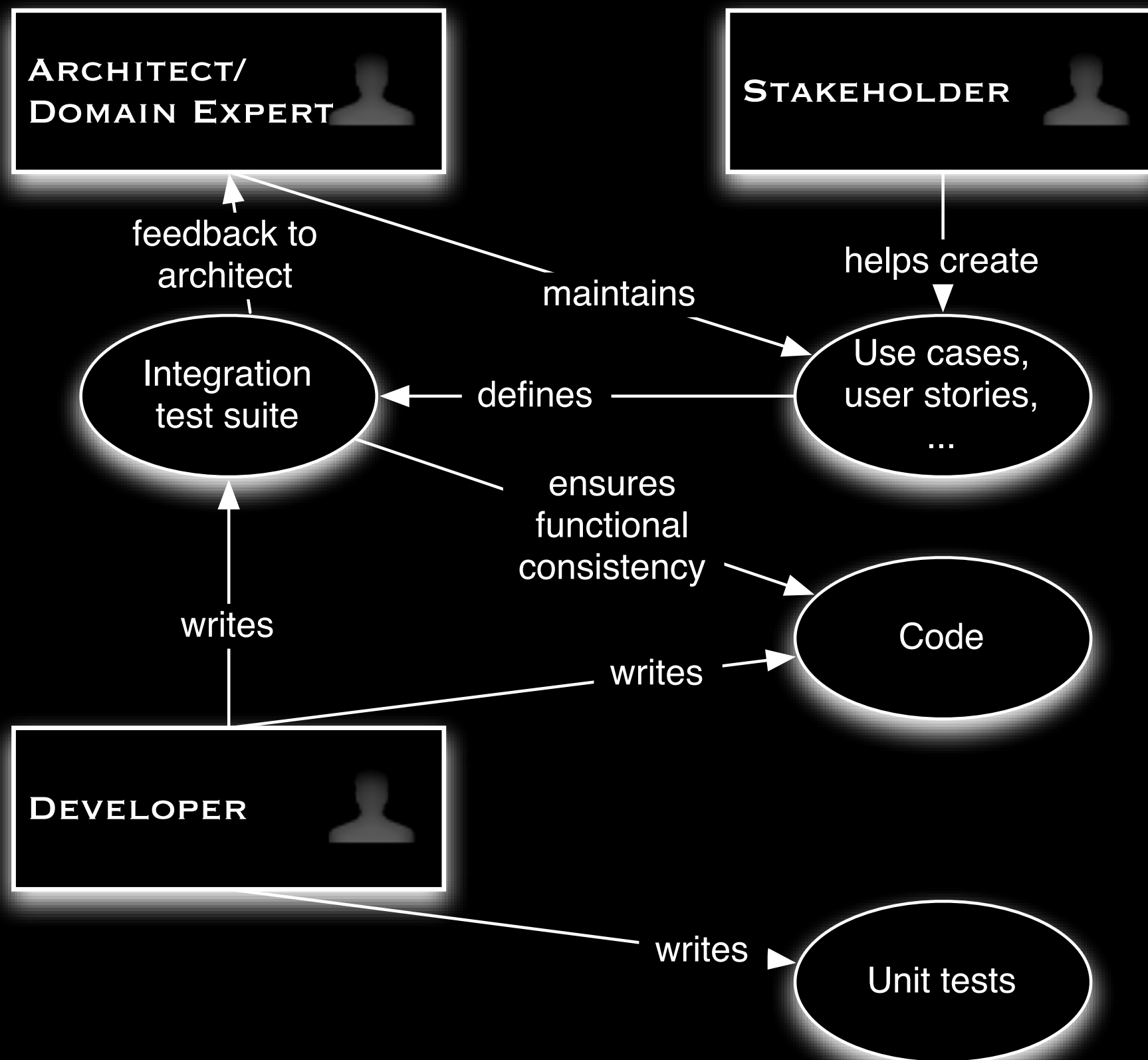


Arbeidsflyt



Overordnet flyt for fangst av funksjonalitet
Hadde det ikke vært en ide om disse menneskene snakket sammen?

Arbeidsflyt



Integrasjonstestene (BDD) kobler prosessene
Feedback til arkitekt – men unittester er der fremdeles
Kundens ønsker vs implementasjon
Trivielt?



Verdens viktigste spørsmål:

“Hva **gjør** denne koden, egentlig?”

“Virker det” god kandidat – forbi det stadiet
Over tid – viktigste spørsmål – nyansatte. Etter 5 år har alle sluttet.
“Tør ikke røre koden!” “Han må ha vært gal!” “Vi begynner på nytt!”
Mulig å gå fra kode til int. test til usecase, som kunden forstår



Konferanser og bøker forenkler
Eksempel på 30 linjer
Bilde: “Dette er en ferdig gjennomtenkt løsning”



Konferanser og bøker forenkler
Eksempel på 30 linjer
Bilde: “Dette er en ferdig gjennomtenkt løsning”



Det guruene ikke sier...

Trege tester

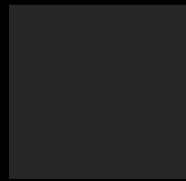
Dokumentasjon og vedlikehold

Det finnes ingen oppskrift på suksess

Treg – maskinvare – alternativ: 10 mann, 3 år, redo from scratch = 30MNOK. Ny laptop=14 dgr
Dok & vedlikehold – må fremdeles ha det med smidige prosesser – nyttebasert
Oppskrift – ingen smørbrødtype som gir garantert suksess – flinke utviklere gjør det
Vellykket utvikling (over mer enn 3 år) er en øvelse i kommunikasjon

Noe å strekke seg etter

Realiteten: Mange delsystemer samarbeider:
periodiske jobber, registrering av kunder,
adminstrasjon/supportapplikasjoner, logistikk,
billing, business intelligence,
sanntidsmonitorering, publisering av
informasjon, mersalg, ...



Eksempler i bøker

Eksempel: Ny cronjobb, arkitektene snakket ikke sammen

Kostnad: "Vi må begynne på nytt" – grunnleggende optimalisering

Spørsmål



Kilder



- **Forsidebilde: Truls Bergskaug**
 - http://flickr.com/photos/t_bergskaug/
- **Motorveibilde fra Wiki commons**
- **Andre bilder: Asle Næss**
 - <http://flickr.com/photos/aness/>
- **Dan North, BDD**
- **Javazone 2007: James Coplien, Five practical solutions to agile myths**