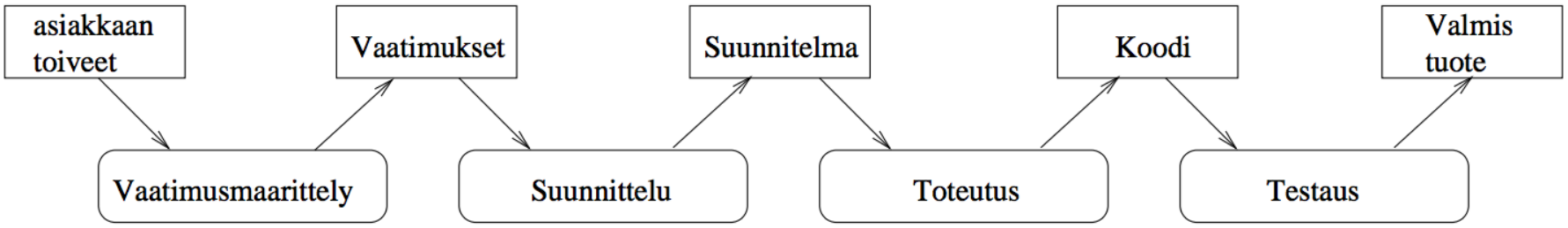
Ohjelmistotuotanto

Matti Luukkainen ja ohjaajat Kalle Ilves, Petri Suhonen, Oskari Nuottonen, Tuukka Puonti

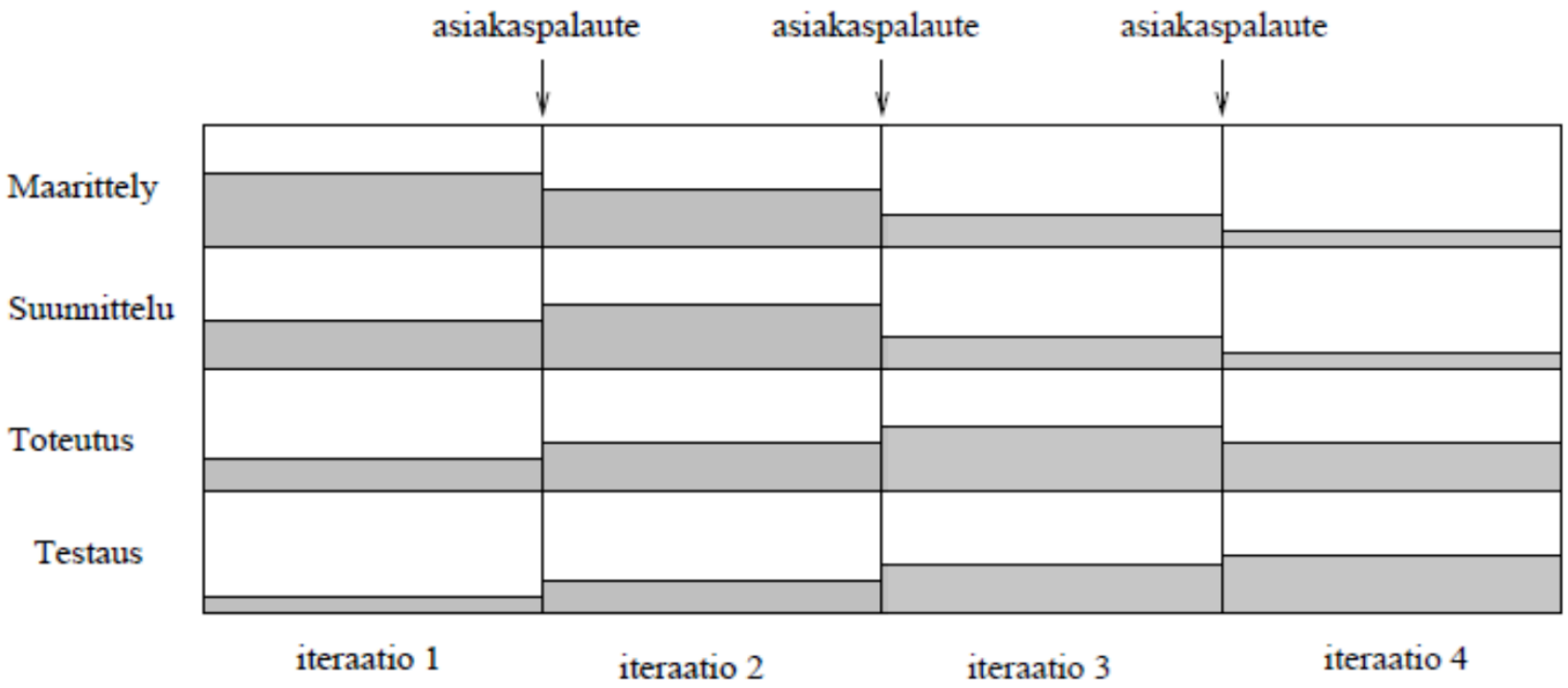
syksy 2022

Vesiputousmalli



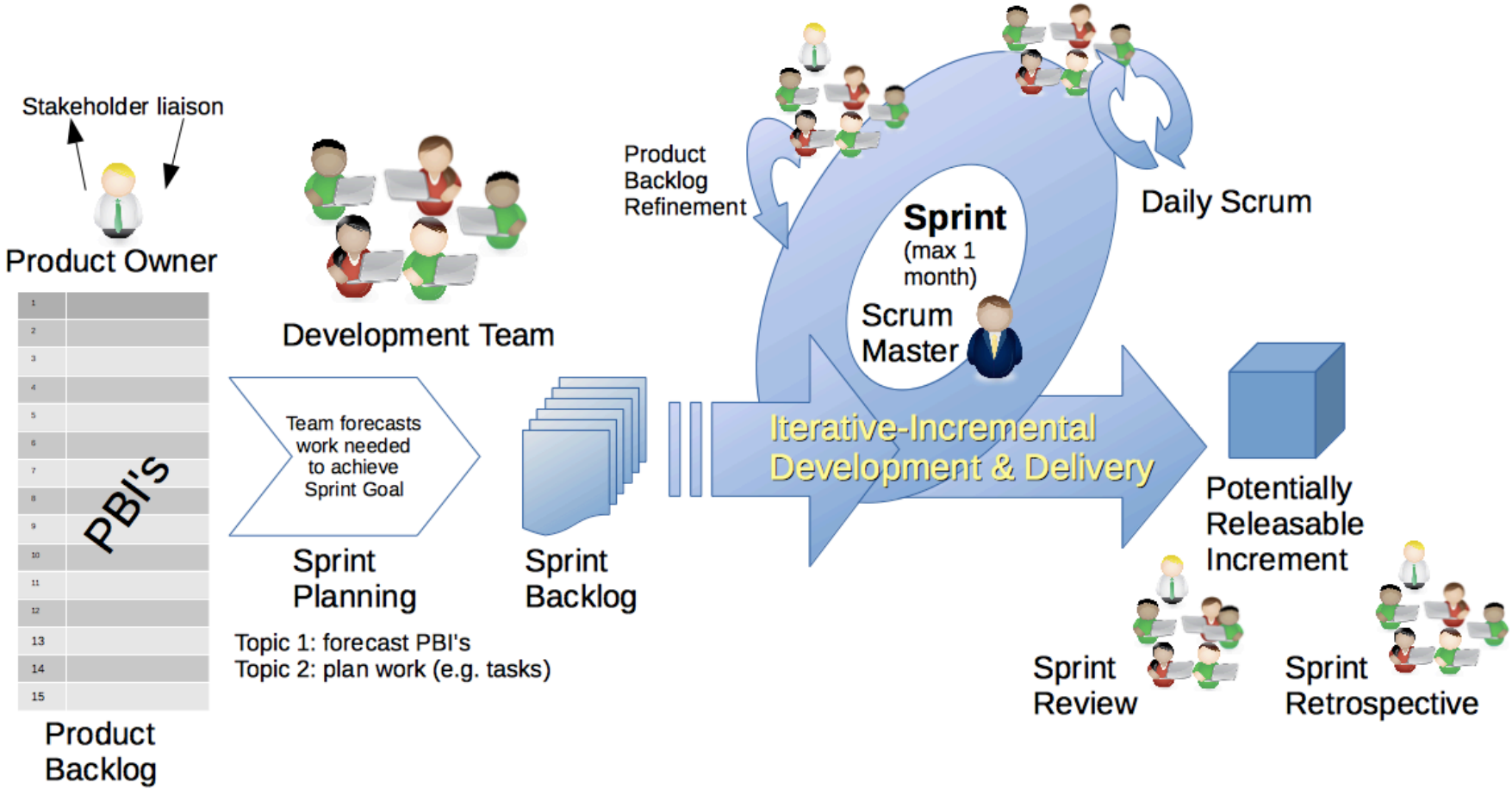
luento 1

Iteratiivinen ohjelmistokehitys



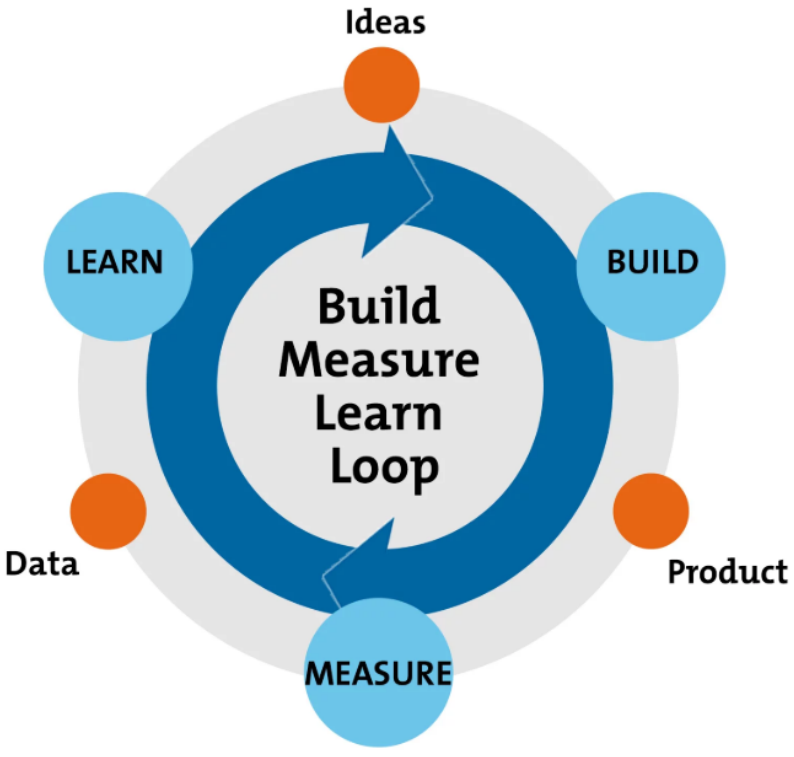
luento 2

Scrum


Stakeholder liaison
Product Owner
PBI's
Product Backlog
Development Team
Team forecasts work needed to achieve Sprint Goal
Sprint Planning
Topic 1: forecast PBI's
Topic 2: plan work (e.g. tasks)
Sprint Backlog
Product Backlog Refinement
Sprint
(max 1 month)
Scrum Master
Daily Scrum
Iterative-Incremental Development & Delivery
Potentially Releasable Increment
Sprint Review
Sprint Retrospective

luento 3

Vaatimusmäärittely 2010-luvulla: Lean startup

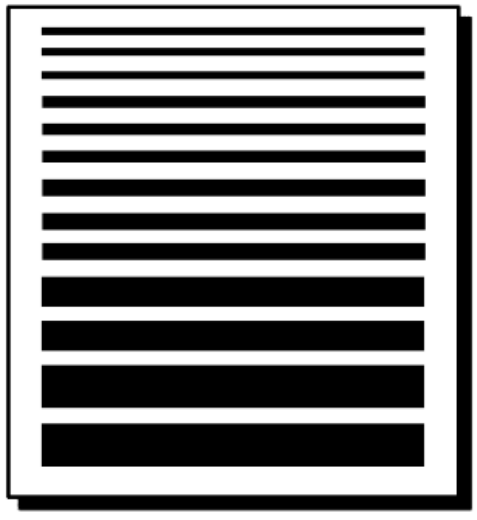


User story

- Mike Cohn:
  - *A user story describes functionality that will be valuable to either user or purchaser of software.*
- User stories are composed of three aspects:
  1. A written description of the story, used for planning and reminder
  2. Conversations about the story to serve to flesh the details of the story
  3. Tests that convey and document details and that will be used to determine that the story is complete

luento 4

High Priority

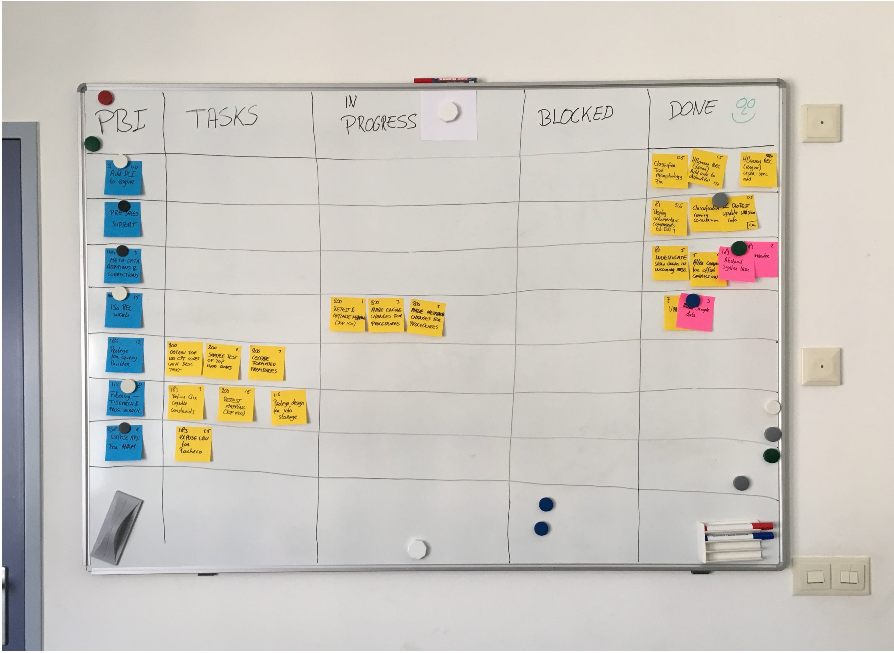


Fine-grained, detailed items ready to be worked on in the next sprint

Low Priority

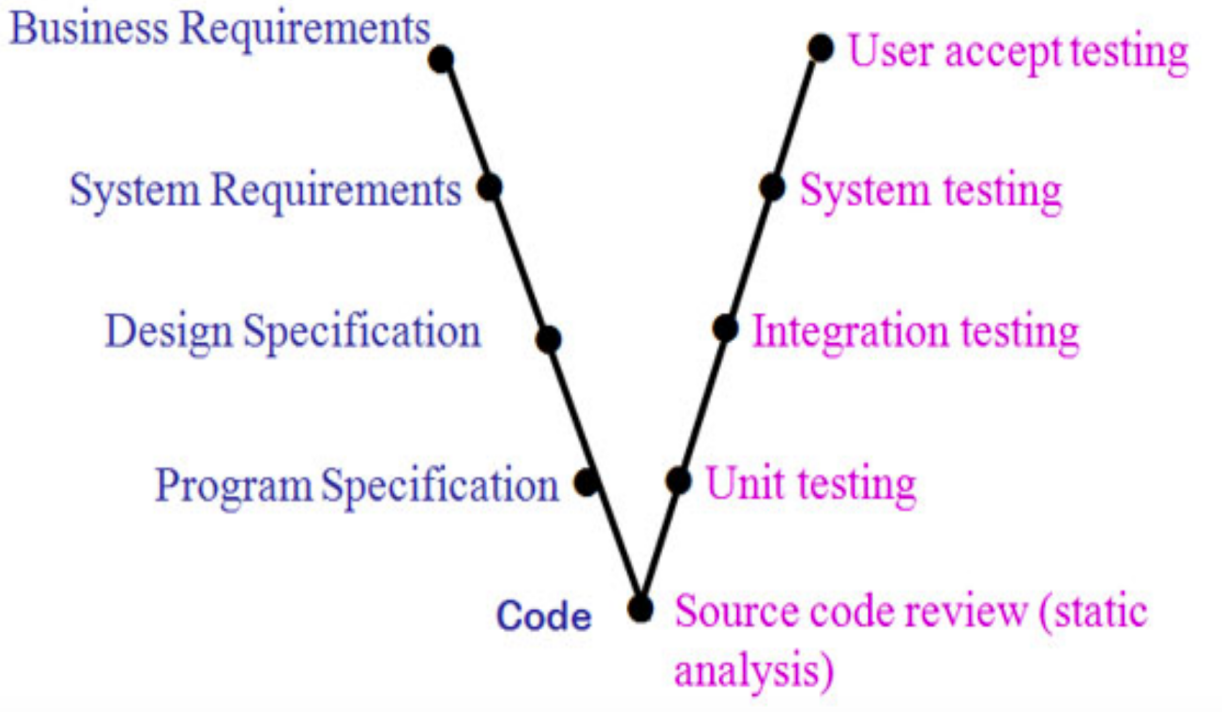
Large, coarse-grained items

WIP-rajoitteet


PBI
TASKS
IN PROGRESS
BLOCKED
DONE
Add DCI to engine
PRE SALES SUPPORT
META-DATA ADDITIONS & CORRECTIONS
ISo BEL work
Prolog for Query Builder
Filtering — TISEARCH & PROG. SEARCH
EXPOSE API FOR MALM
OBTAIN TOP 100 CPT CODES WITH DESC TEXT
SAMPLE TEST OF TOP 1400 CODES
CREATE FORMATED PREMDACES
Define Clix capable constraints
RETEST MAPPING (TOP 1000)
Prolog design for info storage
EXPOSE LBU for Pacheco
RETEST & OPTIMISE MAPPING (TOP 100)
MAKE ENGINE CHANGES FOR PROCEDURES
MAKE METADATA CHANGES FOR PROCEDURES
Classifier Tool Morphology Fix
HBanning REC (frame) Add code to default Bel
HBanning REC (engine) user-spec add
Deploy volumetric components to DA 1
Classification naming conversion
DioTEST update version info
INVESTIGATE SLOW DOWN IN INCOMING MSG.
Alter compo for offset CORRECTION
Abstract Syntax tree
Sample data

luento 5

- *Yksikkötestaus* (unit testing)
- *Integraatiotestaus* (integration testing)
- *Järjestelmätestaus* (system testing)
- *Käyttäjän hyväksymistestaus* (user acceptance testing)

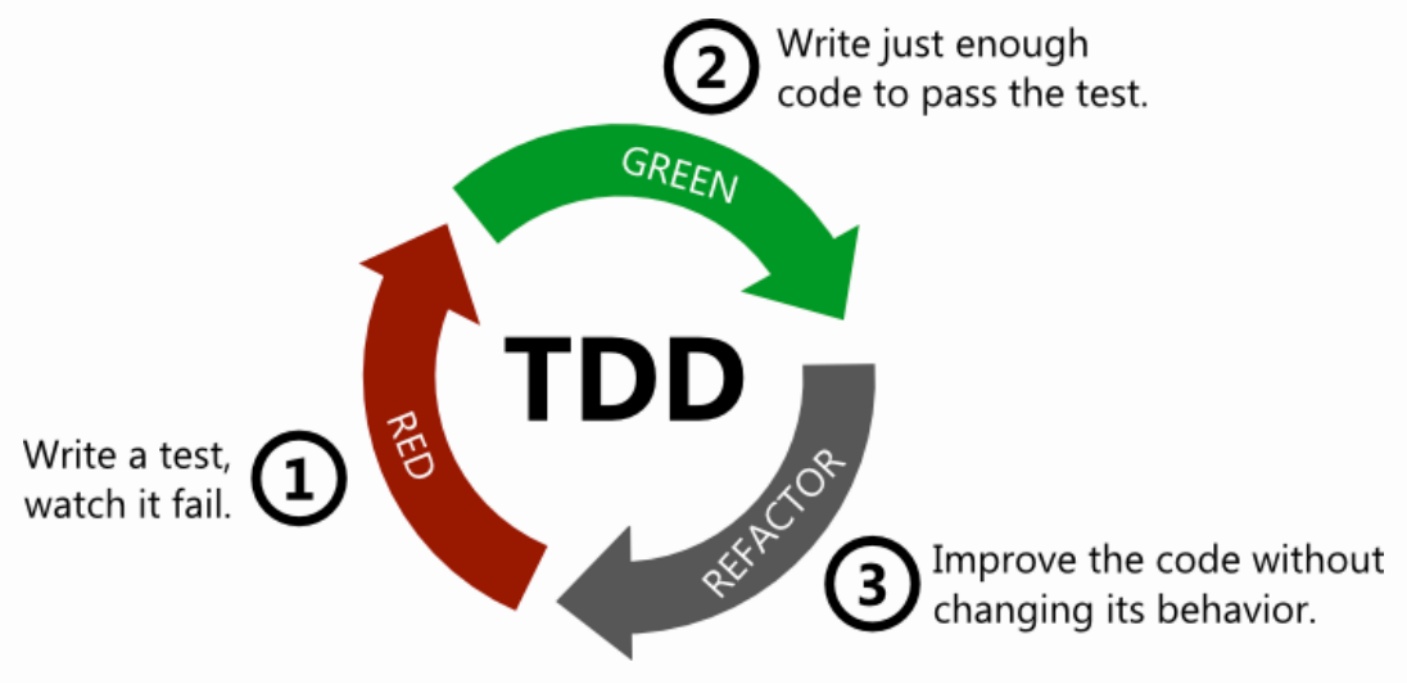


Ohtuvarasto: tyhjä, puolitäysi, täysi

```
class Varasto
    def __init__(self, tilavuus, alku_saldo = 0):
        self.tilavuus = tilavuus
        self.saldo = alkusalto

    def ota_varastosta(self, maara):
        if maara < 0:
            return 0.0

        if maara > self.saldo:
            kaikki_mita_voidaan = self.saldo
            self.saldo = 0.0
            return kaikki_mita_voidaan

        self.saldo = self.saldo - maara
        return maara
```

luento 6

Test driven development (TDD)



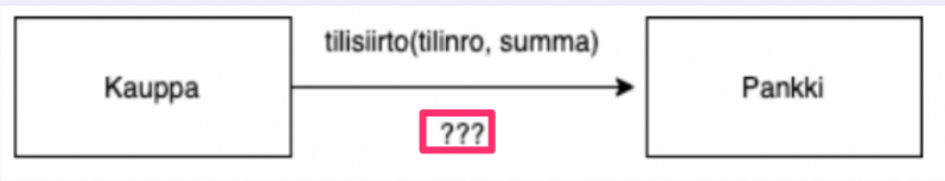
Riippuvuudet testeissä: dependency injection

```
class Laskin:
    def __init__(self, io):
        self._io = io

    def suorita(self):
        while True:
            luku1 = int(self._io.lue("Luku 1:"))

            if luku1 == -9999:
                return

            luku2 = int(self._io.lue("Luku 2:"))

            if luku2 == -9999:
                return

            vastaus = self._laske_summa(luku1, luku2)

            self._io.kirjoita(f"Summa: {vastaus}")

    def _laske_summa(self, luku1, luku2):
        return luku1 + luku2
```

Mock-kirjastot

- Kaupan metodin *maksa* pitää tehdä *tilisiirto* kutsumalla *Pankin* metodia

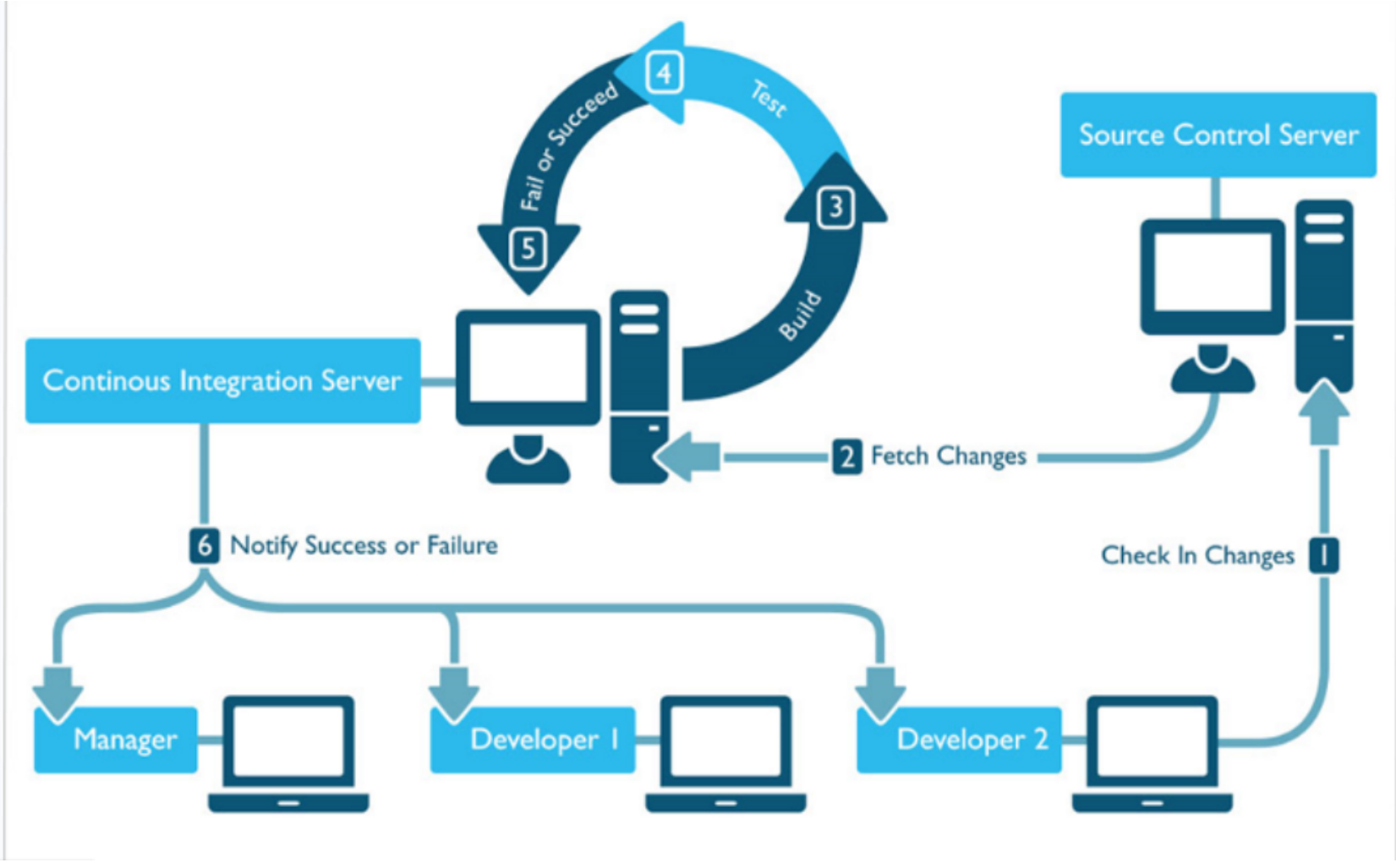
```
my_net_bank = Pankki()
viitteet = Viitegeneraattori()
kauppa = Kauppa(my_net_nank, viitteet)

kauppa.aloita_ostokset()
kauppa.lisaa_ostos(5)
kauppa.lisaa_ostos(7)
kauppa.maksa("1111")
```

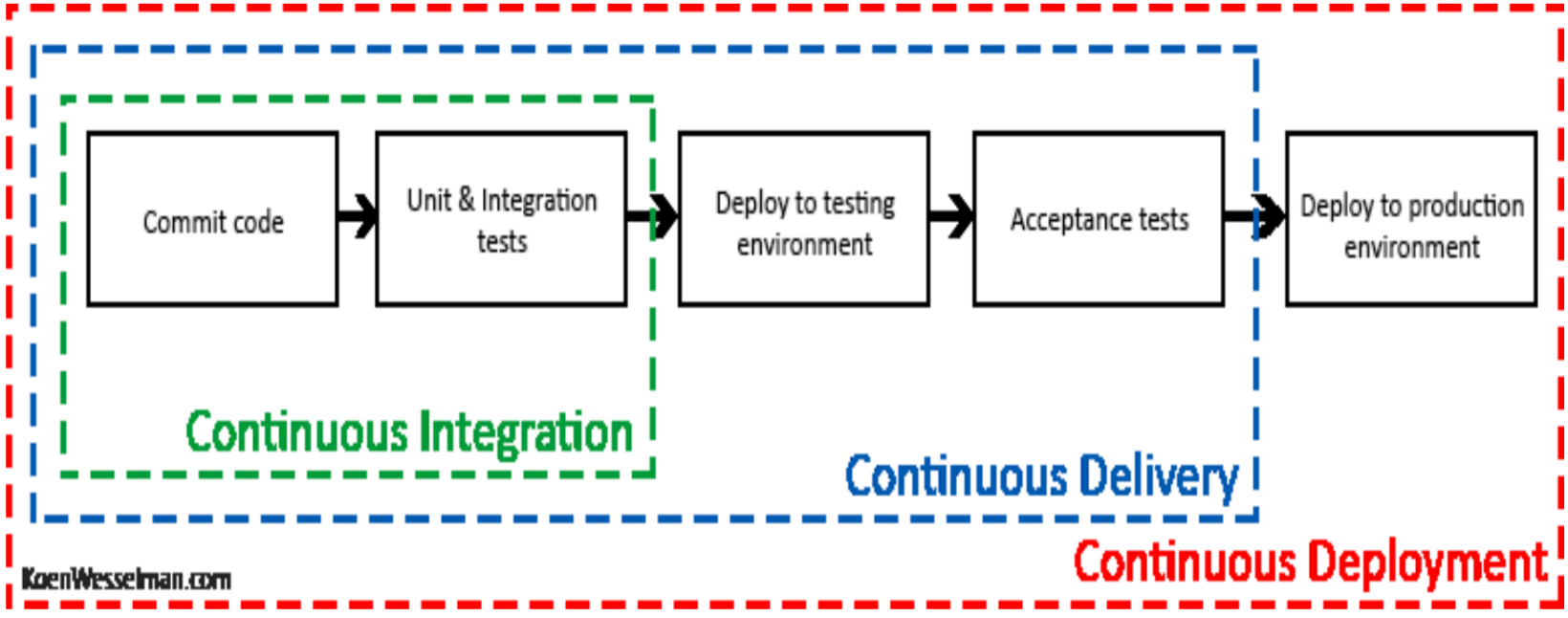


Testit asiakkan kielellä

```
*** Test Cases ***
Register With Valid Username And Password
    Input Credentials  kalle  kalle123
    Output Should Contain  New user registered

Register With Already Taken Username And Valid Password
    Input Credentials  kalle  kalle123
    Input Register Command
    Input Credentials  kalle  foobarfoo1
    Output Should Contain  User with username kalle already exists

Register With Too Short Username And Valid Password
    Input Credentials  k  kalle123
    Output Should Contain  Username too short

Register With Valid Username And Too Short Password
    Input Credentials  kalle  k
    Output Should Contain  Password too short
```

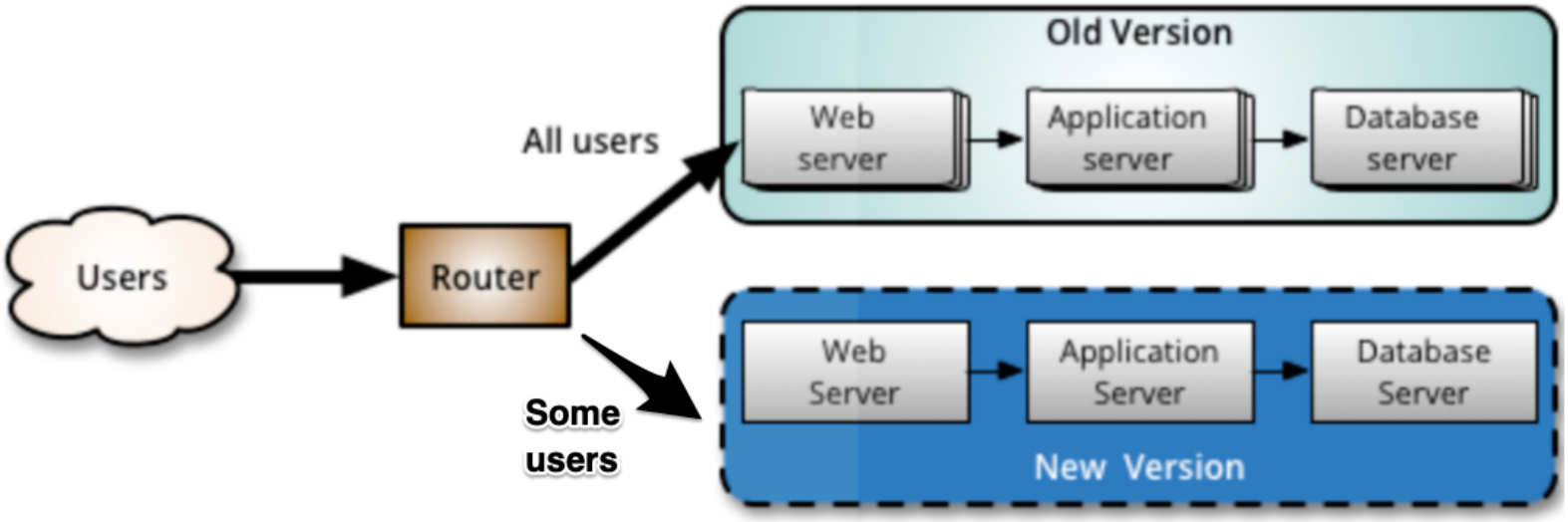
- *Input Credentials*, *Output should contain* ym avainsanoja


Source Control Server
Continous Integration Server
Fail or Succeed
4
Test
3
Build
5
2 Fetch Changes
6 Notify Success or Failure
Check In Changes 1
Manager
Developer 1
Developer 2


Commit code
Unit & Integration tests
Deploy to testing environment
Acceptance tests
Deploy to production environment
Continuous Integration
Continuous Delivery
Continuous Deployment
KoenWesselman.com

Luento 7

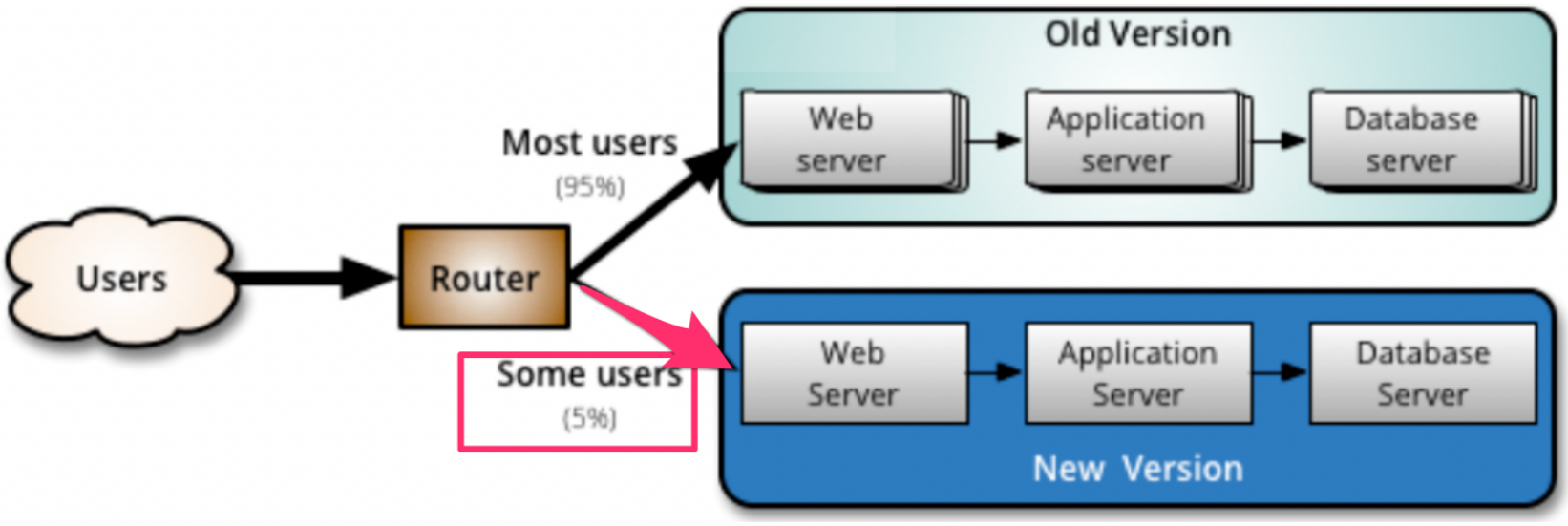
Blue-green-deployment



- Uusi ominaisuus deployataan ensin passiiviseen ympäristöön
- ja sitä testataan
  - osa liikenteestä ohjataan aktiivisen lisäksi passiiviseen ympäristöön ja varmistetaan, että toiminta odotettua

Canary release

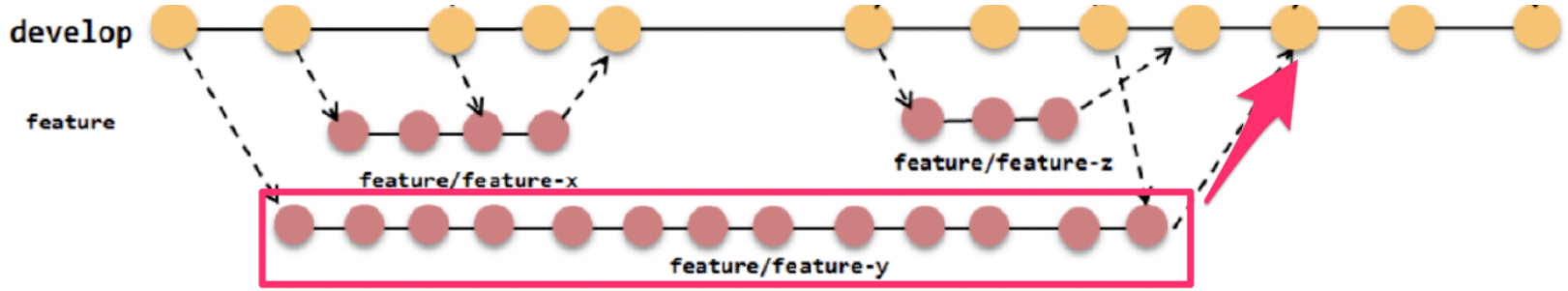
- *Canary-releasessa* uuden ominaisuuden sisältävään ympäristöön ohjataan osa järjestelmän käyttäjistä



- Uuden ominaisuuden sisältämää versiota *monitoroidaan*
  - jos ei ongelmia ohjataan kaikki liikenne uuteen versioon
- Ongelmatilanteissa palautetaan käyttäjät aiempaan, toimivaksi todettuun versioon

Feature branchit

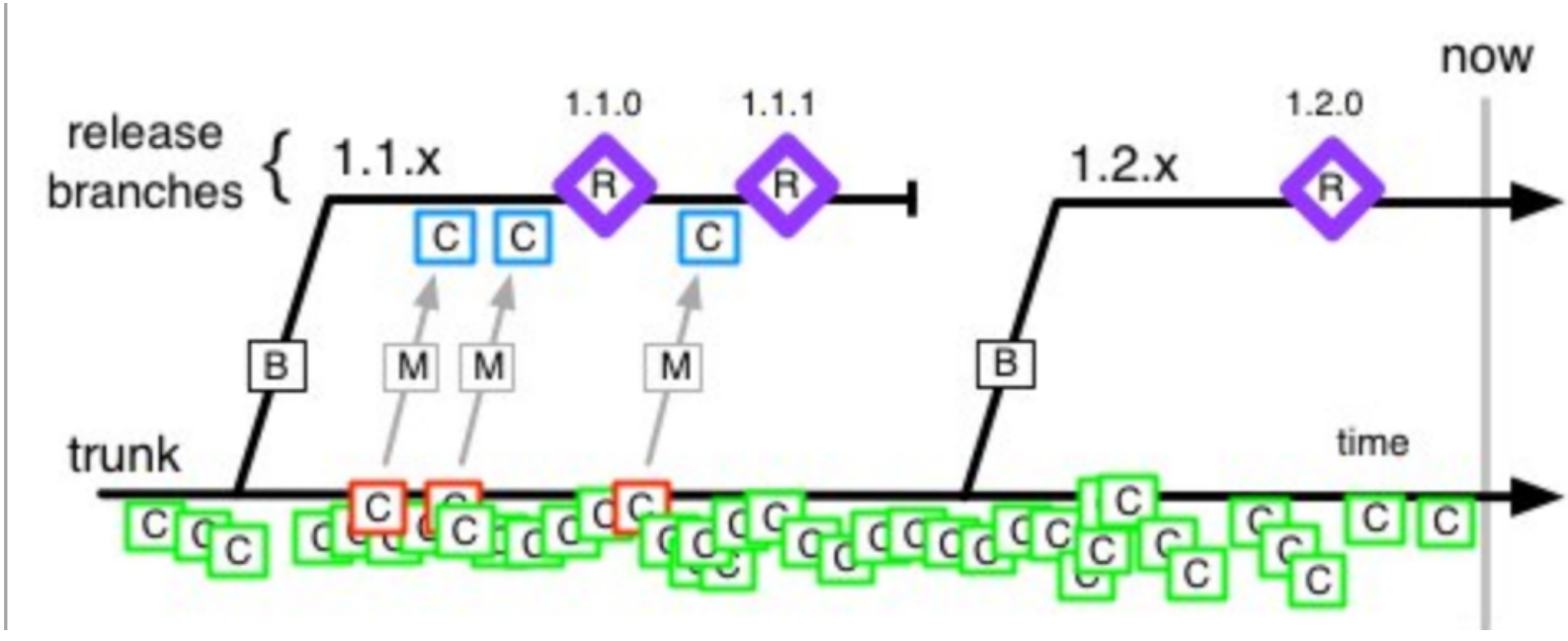
- Uusi ominaisuus, esim. user story toteutetaan ensin omaan versionhallinnan haaraansa



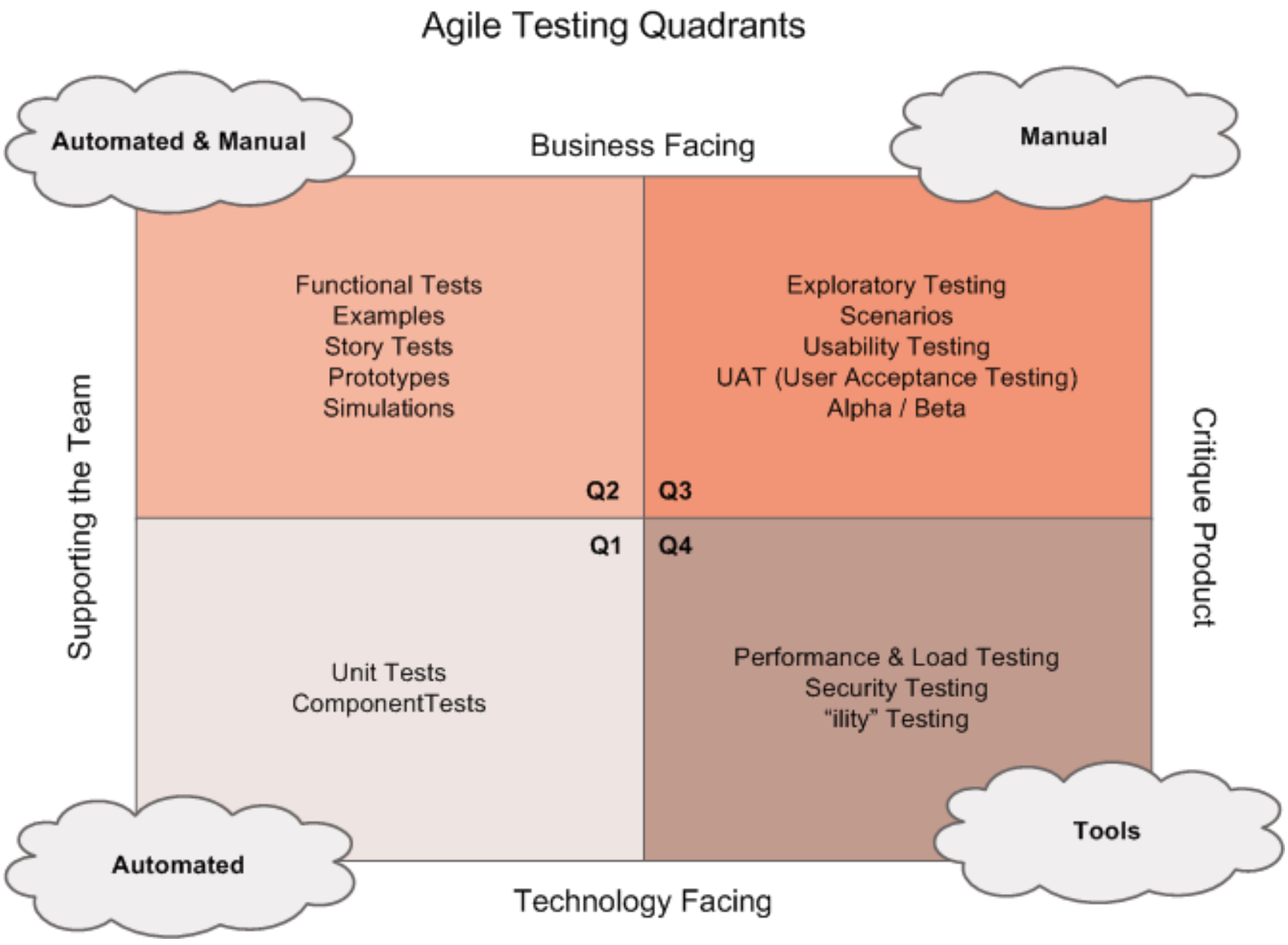
- ja ominaisuuden valmistuttua haara mergetään pääkehityshaaraan

Trunk based development

- Uusi trendi *trunk based development*: pitkäikäisiä feature brancheja ei käytetä ollenkaan
  - Kaikki koodi suoraan pääkehityshaaraan
  - ... josta käytetään nimitystä *trunk*



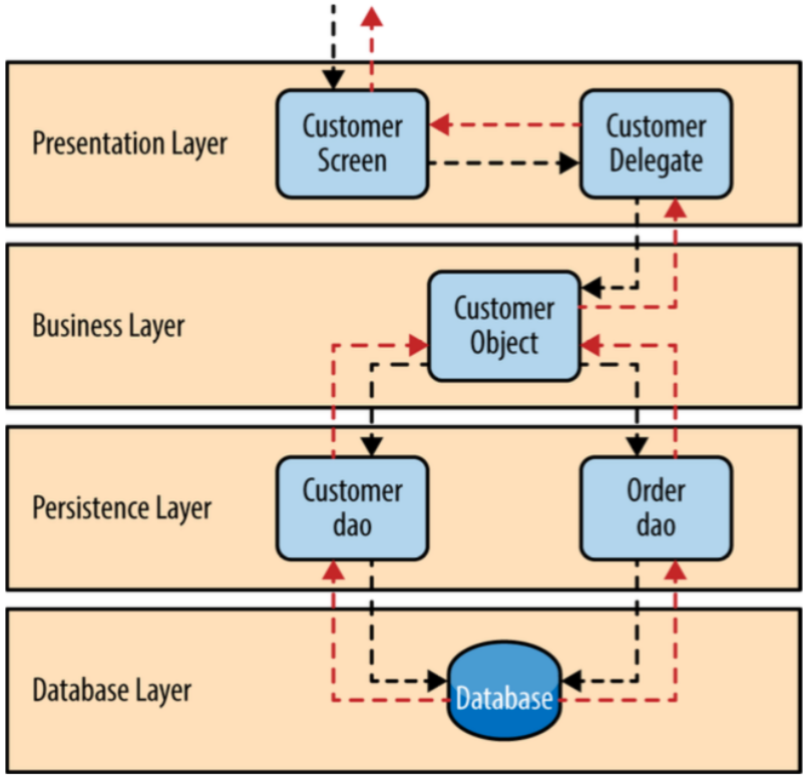
Yhteenveto - ketterän testauksen nelikettä



Luento 8

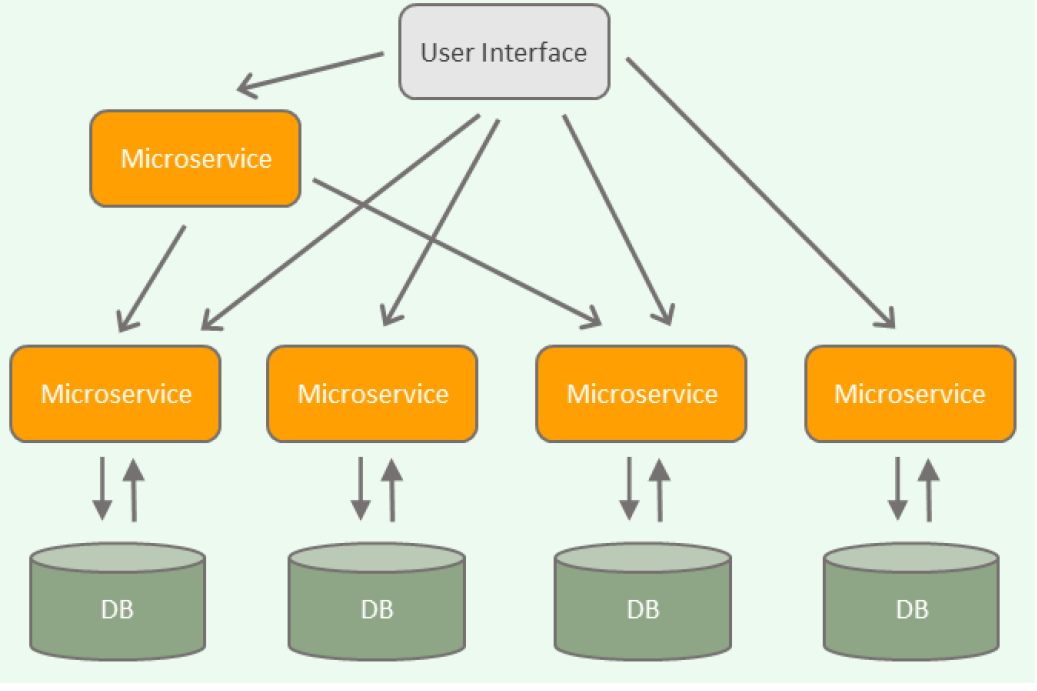
Kerrosarkkitehtuuri

- *Kerros* on kokoelma toisiinsa liittyviä olioita, jotka muodostavat toiminnallisuuden suhteen loogisen kokonaisuuden



Mikropalveluarkkitehtuuri

- sovellus koostataan useista (jopa sadoista) pienistä verkossa toimivista autonomisista palveluista



Laadukas koodi

- Laadukkaalla koodilla joukko yhteneviä ominaisuuksia, tai *laatuattribuutteja*, esim. seuraavat:
  - kapselointi
  - korkea koheesion aste
  - riippuvuuksien vähäisyys
  - toisteettomuus
  - testattavuus
  - selkeys

luento 9

Lean thinking houste