

Objektorienterede metoder



Gang 13. Adrætte processer
Ekstremprogrammering (XP)



projektopgaven i OOM



- Projektvejledning
 - Marcelo Bombaci - holdbarhedsdokumentationen
 - Peter - DIMS (softwaredistributionssystem)
 - Ilsemarie
 - Brian - baseline tool (tjekke versioner af filer)
 - Jan Bjerregaard Madsen - udenlandske vareforsendelser
 - Torben
 - Henrik Klarskov
 - Morten
 - Jens
 - Lisbeth Friis - Luftforureningsvarslingsystem
 - Anders Jørgen - et PoS system
 - Nasser Mobaraki - Andedammens Regnskab
 - Klaus Elmquist - design af procedurale mønstre
 - Anders - **Unittest-framework for X++**



Udviklingsprojekter i virkeligheden



- Tre projekter jeg har været involveret i
 - Optimal udskæring af planker
 - rejseplanen.dk
 - Oprensning af meteorologiske data



Udviklingsprojekter i virkeligheden



- Optimal udskæring af træplanker
 - Del af savskærermaskine
 - Optimeringsopgave
 - 1) Matematisk løsning: Lineær programmering
 - 2) Udvikler prototype i C++
 - Linjestykke-objekter til en kurve, "matematisk API"
 - 3) "joh, men DSPen kan kun tage ANSII-C"
 - 4) Konverterer til "objektorienteret" C-kode (svært læselig)
 - 5) "spild skal - alt andet lige - placeres i for/bag/fordeles ligeligt"
 - Linjestykke-objekter med krumning
 - 6) "Nu er APIet klar - kan du skrive om til at understøtte det?"
 - 7) Etc

- Nye ønsker/krav dukker op løbende - afvisning eller accept?



Udviklingsprojekter i virkeligheden



rejseplanen.dk

- rejseplanen v0
 - Færdigt tysk produkt
 - Modifieret med CGI-script, der modificerede HTMLen
 - Kun station-til-station
- rejseplanen v1
 - Funktioner som v0
 - CGI-script i C++
 - API til motor tilgængeligt fra projektstart
 - OO design + programmering + HTML-skabeloner styret af mig
 - Prøveballon: Ingen krav om design før programmering
 - Andre parter: 1 assistent til API-kodning, 1 projektleder med kundekontakt
- v1 blev en stor succes



Udviklingsprojekter i virkeligheden



- rejseplanen v2
 - Adressefremsøgning (hjemmelavet), kort fra Krak, billetreservation, ...
 - Uklar mudret "magtfordeling" fra start
 - I DSB informatik: konflikt programmør-databasemand
 - Bus-tog-samarbejde hyrede DSB Interaktiv(!) som kunde til DSB Informatik
 - Problematisk kunde-leverandørforhold
 - Kunstig at én virksomhed leger kunde-leverandør
 - Java-servlet til Javaprogram, der kalder "motor" med API (C)
 - "Rigtigt projekt med kunde", prestigefuldt
 - 1) Foranalyse (krav + estimer)
 - 2) Analyse (nye estimer)
 - 3) Design
 - 4) Programmering (nye krav med ønsker om nye estimer dukker op)
 - Ønsker om hvordan adressefremsøgning skulle fungere blev lavet om flere gange af DSB Interaktiv
 - 5) Afprøvning
 - Mange parter, hvem bestemmer?
 - DSB Informatik
 - Programmører (flere og flere som tiden skred frem)
 - Databasefolk
 - Projektledere (skiftende) og ledere
 - Bus-tog-samarbejde
 - DSB Interaktiv: Projektledere (skiftende)
 - HTML-designhus(e!)



Udviklingsprojekter i virkeligheden



- rejseplanen v2 (fortsat)
 - Jeg siger op (med ½ års varsel)
 - Alligevel er ingen klar til at overtage overblikket
 - Driftsproblemer
 -
 - rejseplanen v3 (nuværende)
 - Færdigt tysk produkt
-
- Nye ønsker/krav dukker op løbende - afvisning eller accept?



Udviklingsprojekter i virkeligheden



- Oprensning af meteorologiske data (Cavemet2)
 - Interpolation fra målestationer - nogle af dem upålidelige
 - Beregningskrævende
 - Skulle efterligne Cavemet1 (fortran)
 - Datafiler og noget af GUI er fastlagt
 - Noget af GUI lavet, manglede motor helt
 - Jeg kommer ind som "nødhjælp" 3 måneder før projektafslutning
 - Mange ting uafklarede
 - Oprensningsmetoder
 - Skal nedbør med?
 - Jeg vælger XP-*lignende* metode
 - Jeg: projektleder, designer, underviser, indpisker
 - Parprogrammering
 - Bedre kode
 - Undervisning
 - Overlevering, konsensus
 - Masser af Unit-test

Nye ønsker/krav dukker op løbende - accept!!!



Udviklingsprojekter i virkeligheden



- Oprensning af meteorologiske data (Cavemet2)
- Unit-test-demo: Udregning af funktioner

```
private double udregnFunktion(String navn, double[] fx, double[] fy, double x) {
    int n = 0;
    // søg frem til fx[n] > x
    while (n < fx.length && fx[n] <= x) n++;

    // Tjek grænser
    if (n == fx.length || n == 0) throw new IllegalArgumentException("Funktionen "+navn+" er
ikke defineret for x-værdi "+x);

    // De to nærmeste punkter er n-1 og n. Udregn vægtningsforholdet a
    // sådan at: x=fx[n-1] => a=0 og x=fx[n] => a=1
    double a = (x-fx[n-1]) / (fx[n]-fx[n-1]);

    // interpolér mellem de to nærmeste punkter (fy[n-1] og fy[n])
    double y = (1-a)*fy[n-1] + a*fy[n];

    System.out.println("udregnFunktion: x="+x+" n="+n+" a="+a+" y="+y);
    return y;
}
```



Udviklingsprojekter i virkeligheden



- Skrive testmetoden (OK, jeg gjorde det først bagefter)

```
public void testTjekkerAfNedboer_udregnFunktion()
{
    TjekkerAfNedboer t = new TjekkerAfNedboer();
    double[] fx = { 0, 1, 2, 3, 4, 5, 0, 0, 0, 0};
    double[] fy = { 0, 5, 2, 3, 4, 5, 0, 0, 0, 0};

    assertEquals( 0 , t.udregnFunktion("t",fx,fy,0  ), 0.01);
    assertEquals( 0.5, t.udregnFunktion("t",fx,fy,0.1), 0.01);
    assertEquals( 2.5, t.udregnFunktion("t",fx,fy,0.5), 0.01);
    assertEquals( 5 , t.udregnFunktion("t",fx,fy,1  ), 0.01);
    assertEquals( 2 , t.udregnFunktion("t",fx,fy,2  ), 0.01);
    assertEquals( 3 , t.udregnFunktion("t",fx,fy,3  ), 0.01);
    assertEquals( 3.1, t.udregnFunktion("t",fx,fy,3.1), 0.01);
    assertEquals( 3.6, t.udregnFunktion("t",fx,fy,3.6), 0.01);
    assertEquals( 3.9, t.udregnFunktion("t",fx,fy,3.9), 0.01);
    assertEquals( 4 , t.udregnFunktion("t",fx,fy,4  ), 0.01);
    assertEquals( 4.999 , t.udregnFunktion("t",fx,fy, 4.999  ), 0.01);
    assertEquals( 5 , t.udregnFunktion("t",fx,fy,5  ), 0.01);
    try {
        t.udregnFunktion("t",fx,fy, -0.1 );
        this.fail("bør ikke acceptere -0.1");
    } catch (Exception ex) {}

    try {
        t.udregnFunktion("t",fx,fy, 5.1 );
        this.fail("bør ikke acceptere 5.1");
    } catch (Exception ex) {}
}
```



Udviklingsprojekter i virkeligheden



- Den endelige metode

```
public double udregnFunktion(String navn, double[] fx, double[] fy, double x) {
    int n = 0;
    // søg frem til fx[n] >= x
    while (fx[n] < x && n<fx.length) n++; // -1 && fx[n]<fx[n+1]

    // Tjek grænser
    if (x<fx[0] || n==fx.length) throw new IllegalArgumentException("Funktionen "+navn+"
er ikke defineret for x-værdi "+x);

    // specialtilfælde: Lige på allerførste x-punkt
    if (n==0) n++;

    // De to nærmeste punkter er n-1 og n. Udregn vægtningsforholdet a
    // sådan at: x=fx[n-1] => a=0 og x=fx[n] => a=1
    double a = (x-fx[n-1]) / (fx[n]-fx[n-1]);

    // interpolér mellem de to nærmeste punkter (fy[n-1] og fy[n])
    double y = (1-a)*fy[n-1] + a*fy[n];

    System.out.println("udregnFunktion: x="+x+" n="+n+" a="+a+" y="+y);
    return y;
}
```



Adrætte processer



- Adræt = letvægt
 - Adaptiv - reagere på det der 'dukker op'
 - i modsætning til prediktiv - at planlægge alt på forhånd
 - et radikalt opgør med 'vandfaldsmodellen'
 - Godt til
 - Uklare, delvist ukendte eller skiftende krav
 - Ansvarlige og motiverede udviklere
 - Kunder, der forstår og gerne vil involveres i udviklingsprocessen
 - Dårligt til
 - Udviklingsgrupper med over 100 deltagere
 - Kontrakter med fast defineret pris og funktionalitet
- Manifest for adræt programudvikling
 - **Individuals and interactions** over processes and tools
 - **Working software** over comprehensive documentation
 - **Customer collaboration** over contract negotiation
 - **Responding to change** over following a plan

(citat fra <http://agilemanifesto.org/>)

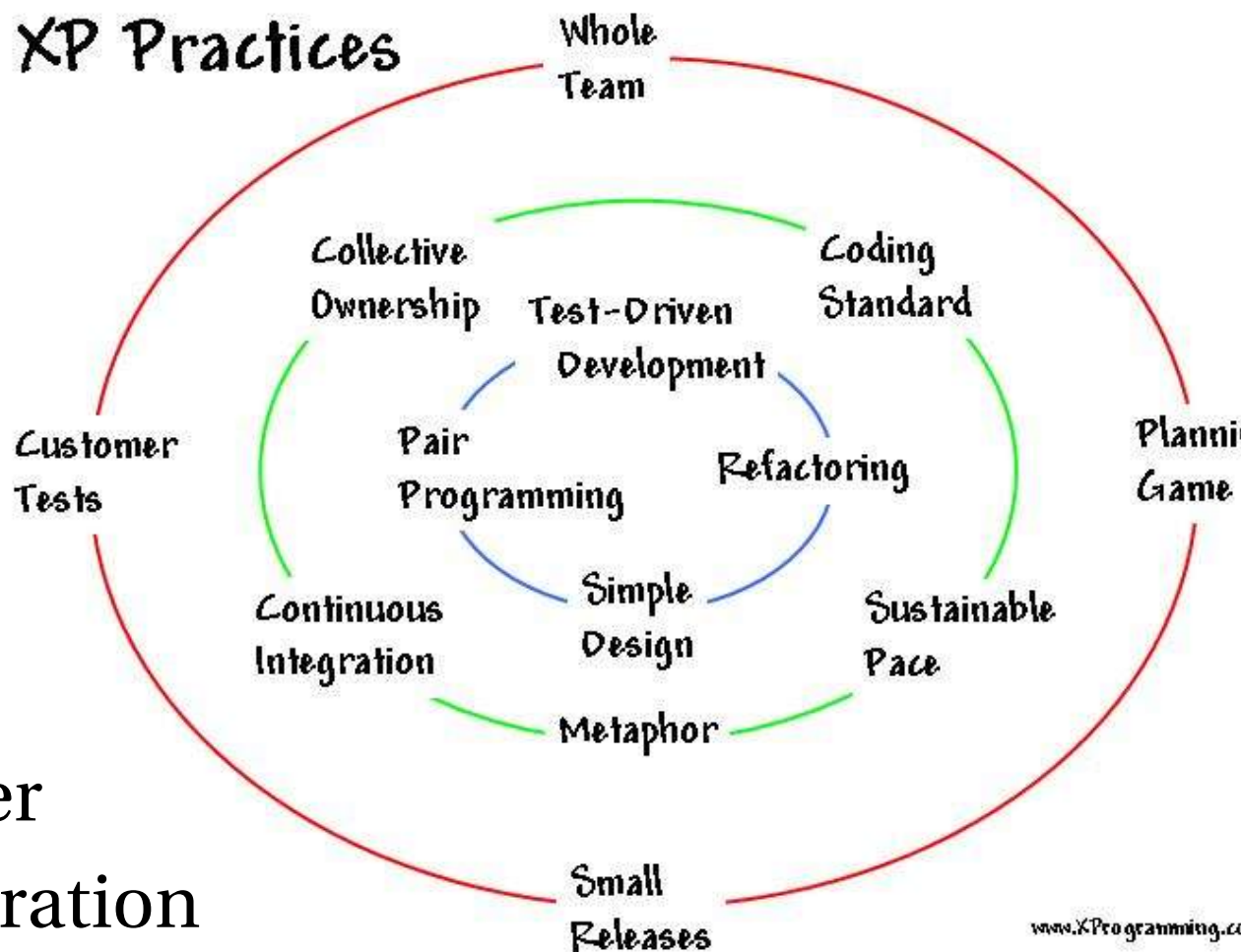


Ekstremprogrammering



- Hyppige afleveringer
- Fælles metafor (sprog)
- Testdrevet udvikling
- Parprogrammering
- Enkelt design
- Refaktorering
- Fælles ejerskab
- Kunden deltager
- 40 timers arbejdsuge
- Fælles kodestandarder
- Løbende systemintegration

XP Practices

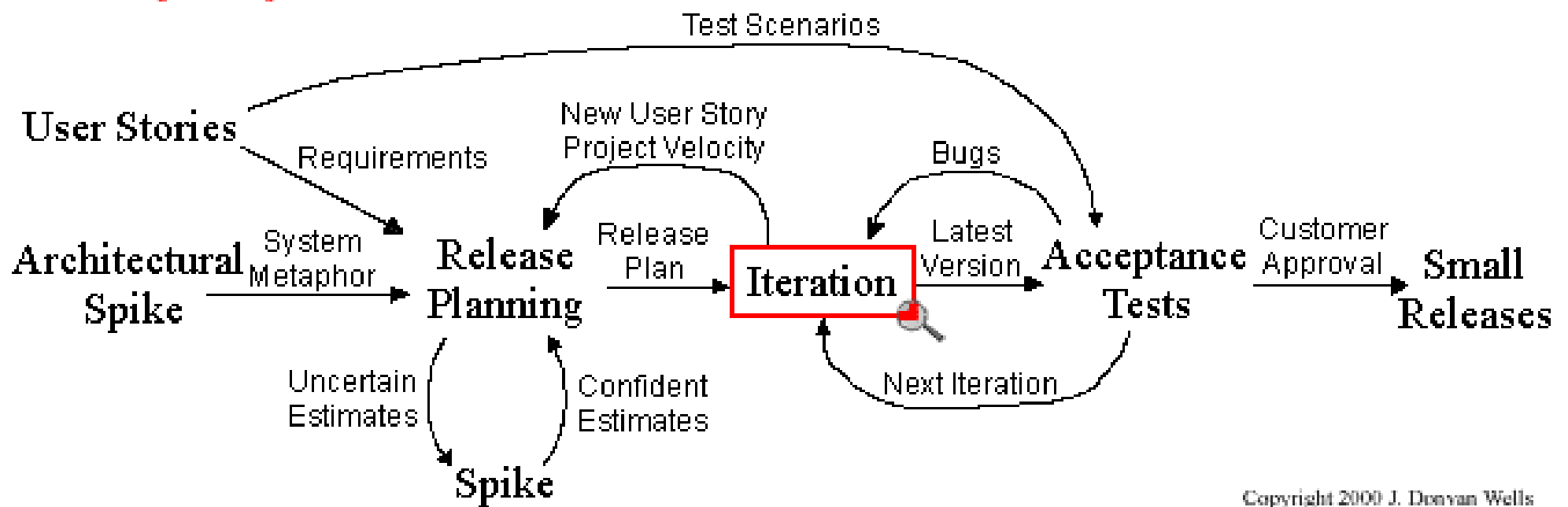




Ekstremprogrammering



Extreme Programming Project



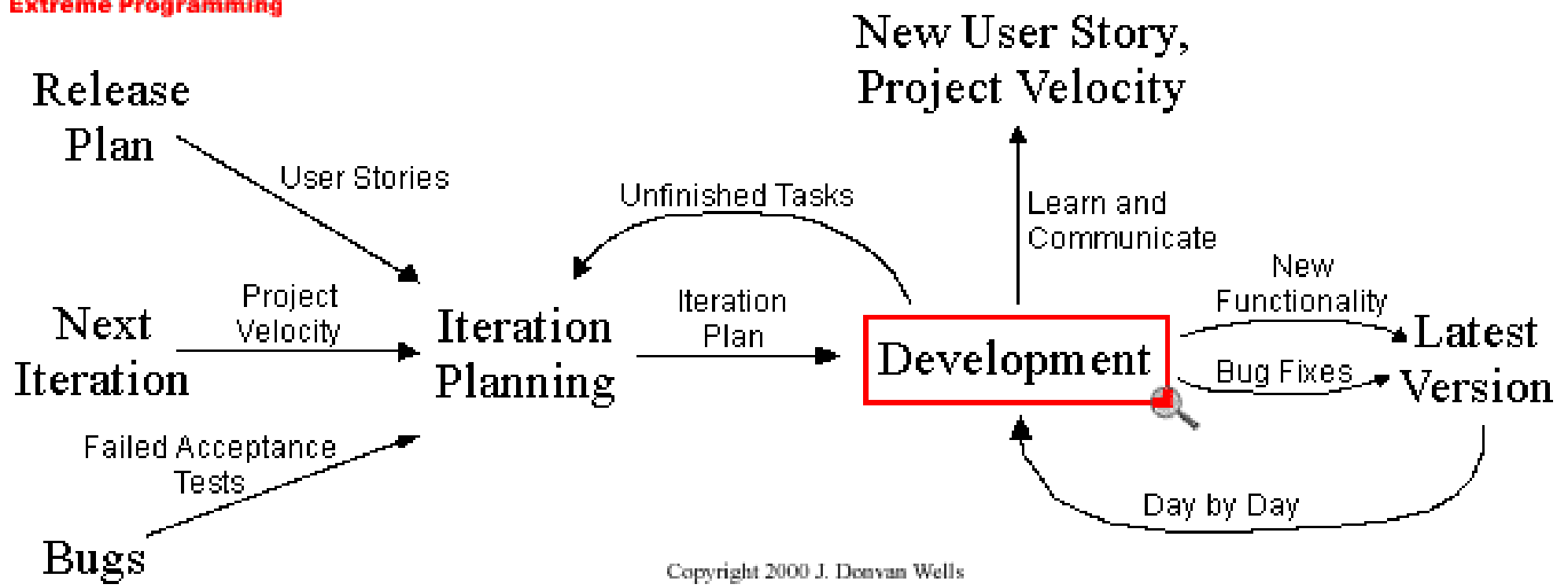


Ekstremprogrammering



Iteration

Zoom Out



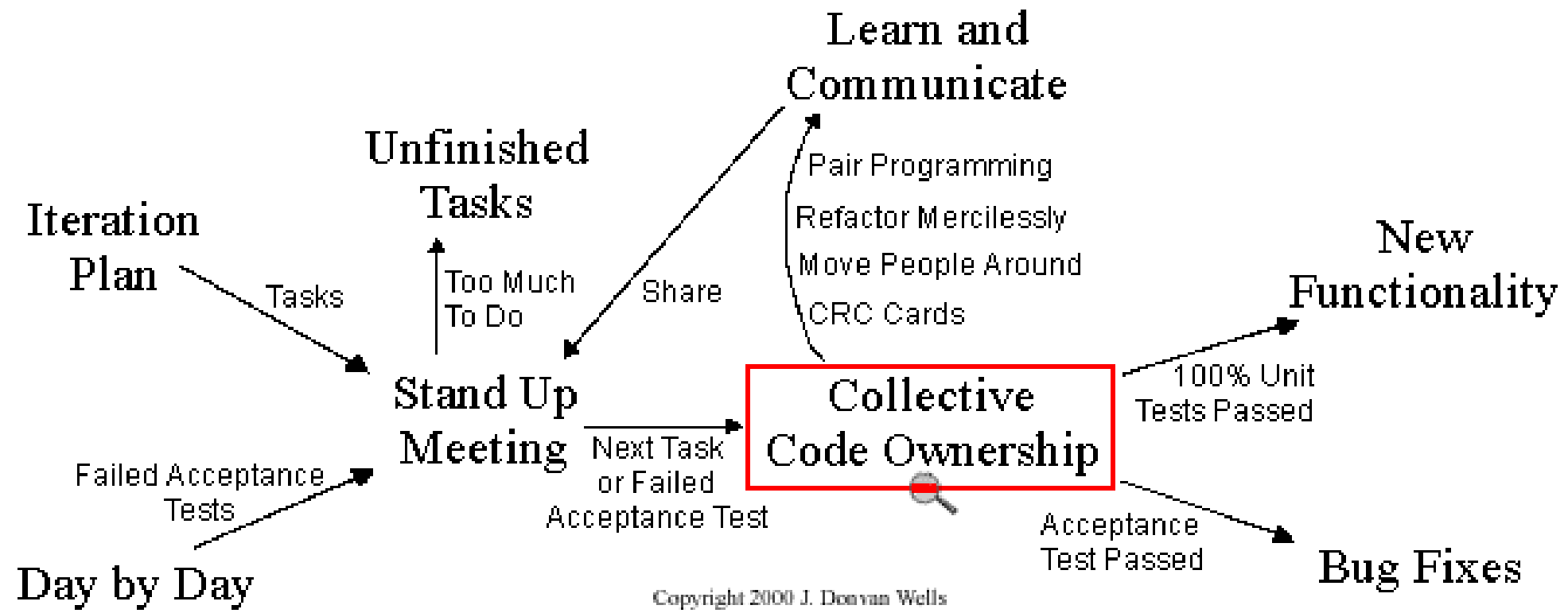


Ekstremprogrammering



Development

Zoom Out



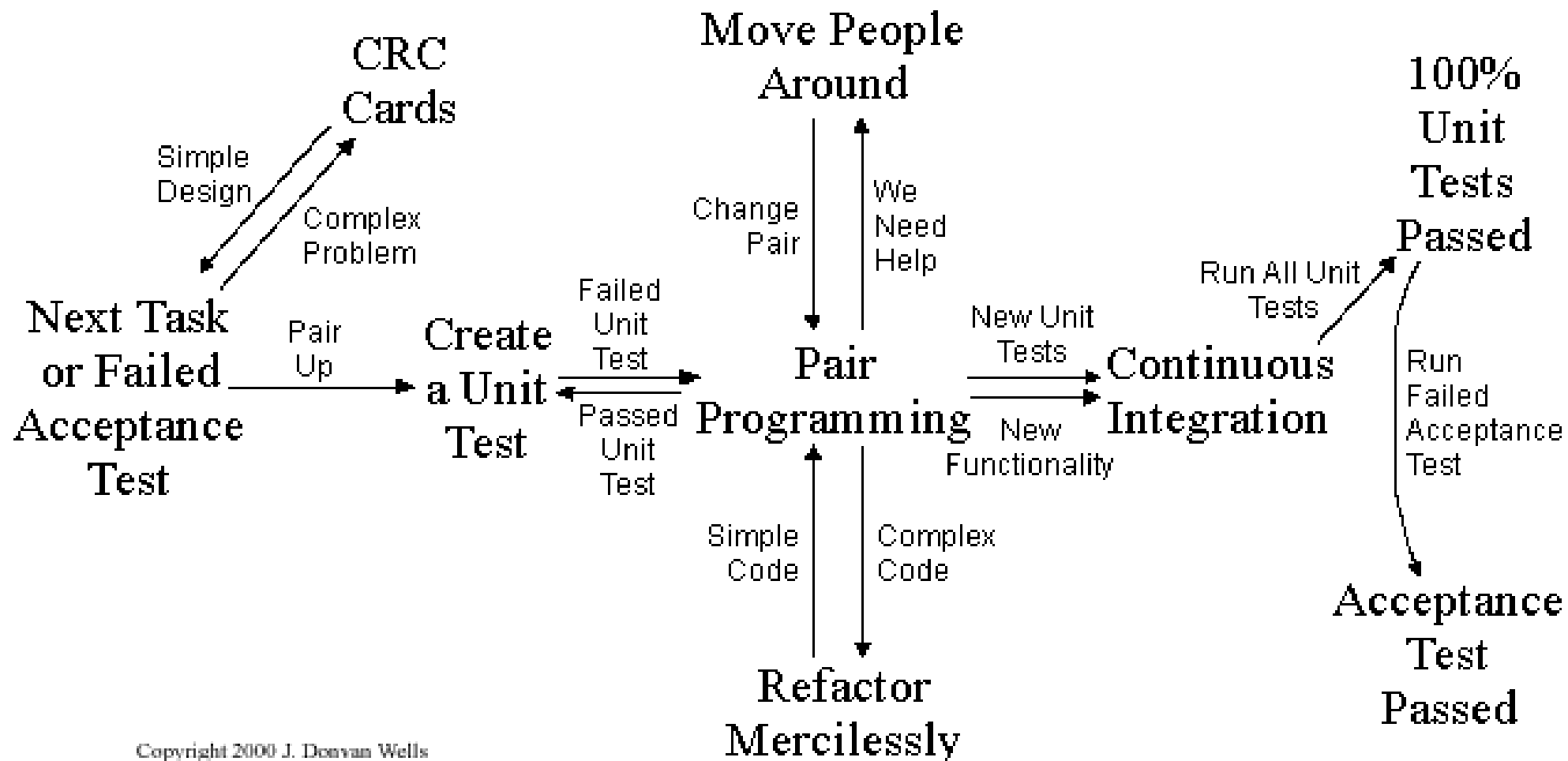


Ekstremprogrammering



Collective Code Ownership

Zoom Out





Adrætte processer



- Læsning
 - Artikel The New Methodology
<http://www.martinfowler.com/articles/newMethodology.htm>
 - Manifesto for Agile Software Development
<http://agilemanifesto.org/>
 - What is Extreme Programming?
<http://xprogramming.com/xpmag/whatisXP.htm>
 - Extreme Programming: A gentle introduction.
<http://extremeprogramming.org/>