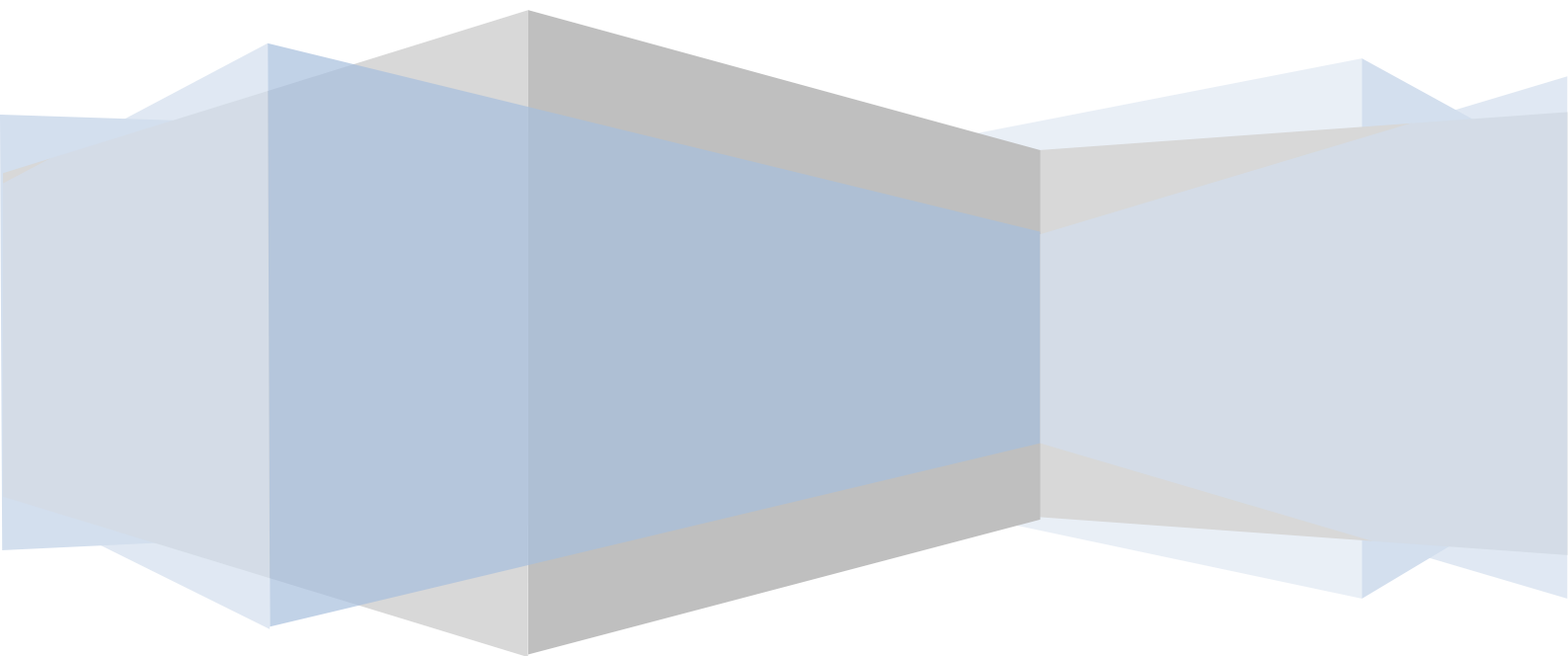


Agile-metoder.dk

World Class Test Management

5. Playing to Win

Poul Staal Vinje



World Class Test Management

Af Poul Staal Vinje

Certified Scrum Master

Certified Scrum Practitioner

poul.vinje@gmail.com

www.agile-metoder.dk

www.softwaretest.dk

Artikel 5 om WCTM: Playing to Win

Indhold

World Class Test Management.....	2
Artikel 5 om WCTM: Playing to Win.....	2
1. Playing to Win	3
2. Tre legepladser	4
3. Scrum	5
4. Extreme Programming (XP)	5
5. Lean	6
6. Kontrolleret offensiv	8

1. Playing to Win

Denne artikel uddyber det fjerde punkt i definitionen på WorldClassTestManagement.

Citat fra den første artikel:

”Playing to Win”, som Kent Beck siger om Extreme Programming. Være opmærksom på Best Practice, og bruge dem helt og fuldt. Ikke kun 10 minutter om dagen, eller når vi er op til det. Men altid være op til det. Sætte sig igennem. Træde i karakter.

Citat slut.

Denne artikel handler om at satse. At turde gøre det man tror på er det rigtige. Både som enkeltperson og som team. For at ”vinde”. Dog peges i slutningen af artiklen på muligheden af risikostyring, samtidig med at der satses.

Men først mere om at satse på at vinde. På ”Wiki” finder man denne tekst:

The idea of Playing to Win is to not be satisfied with merely surviving or playing "not to lose". Go for it. If you don't like your reality, change your reality.

Den beskrivelse er heller ikke dårlig. Det handler altså ikke om kun at overleve. Eller undgå at tabe. Viljen vælger virkeligheden.

I øvrigt hjælper det at agere som vinder. Både som enkeltpersoner og teams. Og både mentalt at tænke som vinder og i sine handlinger.

Hvad vil det så sige at vinde? I en konkurrence er der en taber, når der er en vinder. Men WCTM er ikke en konkurrence. Det kan bedre sammenlignes med en økonomi der ikke fungerer ved at udbytte andre, men ved at skabe merværdi. Skabe en større kage, så alle får et større stykke. Alle kan og skal vinde. Testen bidrager specifikt til produktet når vi tidligt i et forløb kan beskrive testscenarier, acceptkriterier og testcases. WCTM kan ved hjælp af Testdrevet udvikling (TDD) og accepttestdrevet udvikling (ATDD) producere facitlister, der gør udviklingen hurtigere og sikrere.

Når testen starter fra første dag, og er aktiv i hele udviklingsforløbet, skaber den værdi der er større end omkostningen.

På den måde er WCTM med til at hindre fejl i at opstå, snarere end at finde dem når de er begået. Den præventive effekt opstår på to måder. Den ene ved at en potentiel fejl undgås helt. Den anden ved at uklarheder og mangler i kravspecifikationen findes så tidligt, at en række potentielle fejl i koden senere undgås. Det samlede antal fejl der el-

lers ville være blevet begået, reduceres. Princippet gælder ikke kun kravfasen, men også i analyse og designfaser.

Hvis der alligevel opstår fejl satser teamet som minimum på, at fejl findes i den fase de opstår. Fejl må ikke overleve til en efterfølgende fase.

2. Tre legepladser

Der er tre nøgleord til at få luft under vingerne. Det er henholdsvis OMFANG, STYRKE og SATSNING.

Omfang

Første indsatsområde er Kent Becks råd om at gå til yderligheder. At sørge for at de teknikker og metoder vi vælger til udviklings- og testforløbet, bliver brugt på alt, og bliver brugt konstant. To the Extreme. Helt og fuldt. Best Practices er ikke en festtale, med sporadisk brug i praksis. Hvis teamet vælger PairProgramming, TDD eller noget tredje, skal det gøres i stor målestok. Måske ikke 100%, fordi det ikke er et spørgsmål om at være dogmatisk. Men det skal gøres så det kan mærkes at det er dét der er standarden. Så det bliver signifikant. Så det præger forløbet. Man kan ikke sige at man har fået erfaringer med noget, ved at arrangere et eksperiment på et par timer. En afprøvning af TDD kræver at det bruges helt og fuldt i en hel iteration, omfattende hele funktionaliteten i iterationen. Først da har man haft en afprøvning der kan bruges som erfaringsgrundlag.

Styrke

Desuden skal det ske med saft og kraft. Så det kan ses og høres. Hvis et team vælger TDD, så skal flest mulige involveres, både udviklere, testere og brugere. Og der skal tales om det, med de begreber der hører til. Det er bedre at arrangere TDD sessioner som workshops med resultaterne på en Whiteboard, end at overlade det til teammedlemmerne at få det gjort. Implementering handler om at sætte det håndfast på skinner, så det høres og føles. Nye tiltag, der kryber langs panelerne, er dømt til at dø en stille død.

Satsning

Test Manager skal turde turde. Men skal man satse hele bagerbutikken? Uden et sikkerhedsnet. Hvis der er valgt TDD kan teamet så roligt droppe de afsluttende testfaser?

Det er måske sikrere at bruge nogle af de testtimer der er til rådighed, på en afsluttende traditionel test? At vente til udviklerne har sagt de er klar til release, og så bruge testtimerne på at finde fejl?

Men hvis man beholder en afsluttende testfase, hvordan kan man så tage TDD alvorligt. Og regne med at alle går helhjertet ind for det.

Det vil altid være en udfordring at satse. Men det er et problem hvis det er tydeligt at man sikrer sig ved at have alle de kendte mekanismer på plads, samtidig med at man tager de nye i brug.

3. Scrum

Hvilke dele af Scrum er vigtig for Playing to Win? Det helt afgørende element er princippet om den selvstyrende gruppe. Organisationen skal turde den. Teamet skal efterfølgende ville selvlædelsen helhjertet. Tage ansvaret for alt, både udvikling og test. Plus det styringsmæssige, herunder ansvaret for at leve op til alle aftaler.

Det er teamets fælles ansvar at der ikke leveres fejl i produktet, og ikke mindst ved hjælp af præventive tiltag, for at hindre fejl i at opstå overhovedet.

Samarbejdet med produktejeren er også et proaktivt middel. Produktejeren kan bede teammedlemmer om tidligt at opstille acceptkriterier og testscenarier for hvert krav. Teammedlemmer kan alternativt invitere sig selv til at gøre det. Bruge test som specifikation.

4. Extreme Programming (XP)

Disse dele af XP er særlig vigtige for Playing to Win:

- Collective Ownership
- Continuous Integration
- Pair Programming
- Test Driven Development

Det er dele der bryder med en traditionel arbejdsdeling. Samtidig er det dele der bryder med traditionelle processer, både vandfald, V-model og W-model. Koden er ejet af alle. Hele koden. Den er skrevet af skiftende par.

Der er en XP disciplin mere der fortjener omtale:

- Sustainable Pace

Den svarer til Lean's Mudi, og er princippet om at undgå pres og stress.

XP ikke bare satsning. Det er i virkeligheden også sikkerhedsnettet. De 13 discipliner giver fundamentet for at satse. Satsning og forankring går i dette tilfælde hånd i hånd.

5. Lean

I Lean Software Development udføres kun værdifulde aktiviteter. De defineres med tre karakteristika:

- Kunden ønsker at betale for dem
- De kan udføres rigtigt første gang
- De transformerer produktet og tilfører det værdi

Teamet skal have viljen og modet til at droppe alle aktiviteter der ikke skaber værdi. Og viljen og modet til at stoppe arbejdet hvis de finder en fejl. Årsagen til fejlen skal afdekkes, og teamet skal forbedre sin proces for at undgå at denne type fejl opstår igen.

Flow

Sats på at få produktejeren involveret. Ekstremt meget involveret. Hold fast i at færdiggøre en del ad gangen. Bed produktejeren om at acceptteste under vejs. Ikke bare for at få en mening eller opfattelse, men for at få en reel godkendelse.

Flowet er også med til at begrænse antallet af handovers.

Kaizen

Kaizen sørger for at teamet holder fokus. Teamet satser på konstant forbedring i hele udviklingsforløbet.

Vær ærlig på møderne. Stil spørgsmålet: Holder vi en testgæld på 0 (nul)? Og gør vi noget ved det, hvis gælden er for høj?

Arbejd løbende med at forbedre alle tre indsatsområder: Omfang, styrke og satsning.

Built-in Quality

Skab det rette produkt under vejs. Kvalitet kan ikke testes ind i produktet.

Ingen afsluttende test? Ingen fejl rapporter? Utopisk? Men princippet er værd at forfølge. Unittest er en del af konstruktion og der skal satses på en 100% dækkende unittest. Det taget i hvert fald toppen af fejlene. Sats som tidligere nævnt på at fejl som minimum findes og fjernes i den udviklingsfase de opstår.

Accepttesten skal udføres løbende. Det er her produktejeren kommer ind i billedet. Det handler både om kvalitetsstyring af kravene ved hjælp af reviews og inspektioner, og om testdækning af kravene med scenarier og acceptkriterier.

Udviklernes TDD giver også en løbende test.

Muda

Lean arbejder med at opstille en værdikæde der samtidig kan bruges til at identificere og eliminere spildtid.

Playing to Win indebærer at spild, det vil sige ikke-værdiskabende aktiviteter, droppes. Brug artikel 3 i denne serie som inspiration.

Det er værd at bruge tid på at fjerne spildtid, selvom det i sig selv er Muda.

Muri

Der er følsomme emner i dette område. Når artefakter svinger i kvalitet. Hvad skal teamet gøre når nogen i teamet producerer kode der er under den aftalte standard? Måske ikke opfylder Definition of Done. Det samme med design og test artefakterne.

Et team der går efter at vinde, tager det op. Både peer-to-peer og på Kaizen møder.

Teamet skal ikke acceptere Technical Debt i form af en svingende kvalitet af det fysiske system. Der må ikke tænkes kortsigtet, for blot at overleve release. Tænk langsigtet på at produktet skal vedligeholdes og videreudvikles.

Mudi

Sats på at det ikke er nødvendigt at presse og stresse. Hvis opfølgningen viser at teamet er bagud, er svaret at fjerne nogle opgaver, for at nå deadline. Svaret er ikke overarbejde, eller at sætte flere ressourcer ind.

Agile Test Management beskæftiger sig både med kvaliteten af produktet og processen, og til den sidste del er Lean's tanker om Mudi helt på sin plads. Det er det samme område der adresseres af XP's Sustainable Pace.

6. Kontrolleret offensiv

Hvis man spiller for at vinde, og ikke bare undgå at tabe, skal både enkeltpersoner og teamet sætte handling bag ordene. Der skal presses højt oppe på banen for at testen ikke skal ende som en ren defensiv, hvor bolden blot sparkes væk.

Det kan dog være en kontrolleret offensiv, med mulighed for hurtig omstilling til defensiv. Eller på lidt mere direkte dansk: teamet kan vælge lidt risikoanalyse.

Figuren nedenfor viser rammerne til en risikoanalyse. Den første kolonne viser nogle eksempler på tiltag et team har valgt. Risikokolonnen dokumenterer nogle risici. Den sidste kolonne foreslår nogle tiltag til at imødegå truslerne. Det er en god ide at snakke det igennem i et team der satser på Agile Test Management.

Eksempel på kontrolleret offensiv:

Tiltaget	Risiko ved Playing to Win	Kontrolleret defensiv
Built-in Quality.	Det betyder at vi arbejder med løbende test, lav testgæld, og ingen afsluttende testfaser. Vi er måske ikke vant til at arbejde på den måde.	Automatiseret regressionstest. Den bruges under vejs, men kan også fungere som afsluttende test.
TDD	Der slipper måske fejl igennem, fordi udviklerne ikke er trænet i unittest.	Udviklerne trænes i unittest. Stil tydelige, håndfaste og højlydte krav om brug af xUnit værktøjer.
ATDD	Produktejeren tager det ikke alvorligt, i samme grad som en traditionel bruger- og accept-test.	Teamet skal støtte og hjælpe produkt ejeren med at opstille acceptkriterier og testscenarier, allerede mens Userstories befinder sig i Produktbacklog, og i god tid før de udvælges til en iteration.
PairProgramming	Udviklerne er ikke altid motiverede.	De mest anerkendte udviklere går forrest. Bevidst brug af rollemodeller og opinionsdannere.
Ingen stress og pres	Der er måske noget funktionalitet vi ikke når at levere.	En aftale om at det lavest prioriterede kan fjernes.