

## Personal Extreme Programming

- agil metode for single udviklere.



*Scrum, XP og andre agile metoder anvendes i dag på langt de fleste udviklingsprojekter. Men der er én type projekter, som er ladt lidt i stikken. Det er de små udviklingsopgaver, hvor kun en person står for processen - hvor parprogrammering, teamsamarbejde ikke er en mulighed. Ofte tyer udviklerne til at bruge de populære agile metode, og blot undlade at bruge de elementer fra metoden, som kræver samarbejde. Men det går ikke! For pludselig står man i en situation, hvor beskrivelsen af software kun er tilstede inde i udviklerens hoved, og der ikke er gjort nogen foranstaltninger til at sikre kvalitet og overdragelse af viden til fremtiden.*

### Agil software udvikling.

Programmører, som har arbejdet med for programmering i en årrække, har med sikkerhed bemærket at en revolution er skyllet ind over softwarebranchen. Der er sket store forandringer i de softwareudviklingsmetoder vi anvender. Med det agile manifest kom kollektiv ejerskab, teams,

og omstillingsparathed øverst i programmørens værktøjskasse. Med ét vægtede samarbejde med kunden og udviklingsgruppen højere end omfattende dokumentation, og væk var mantraet om at man ikke kunne begynde at kode, før alt papirarbejdet var gjort.

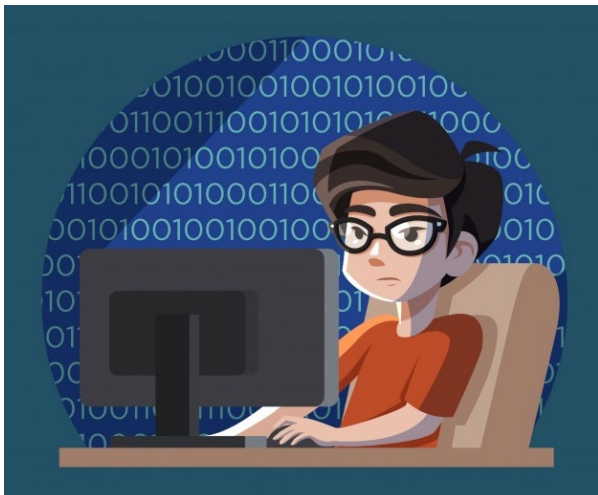
### Extreme Programming.

En af de mest populære agile metoder er Extreme Programming (XP), som bygger på arbejde i teams. Hurtige og små releases, parprogrammering og tæt samarbejde mellem kunden og udviklingsteamet er nogle af de principper branchen har taget til sig. I alt er der 12 principper eller værktøjer der ligger til grund for XP, og som sikrer den fornødne kvalitet i programmer. Principperne er kort gennemgået her

- System Metaphor. En vision for det system som man er ved at konstruere
- Small Releases. Du indfører ganske få nye features i hver version.
- Continuous Integration. Integrere koden ned i den fælles "main branch", og frigive nye versioner ofte - måske dagligt.
- Simple Design - hold systemdesignet enkelt.
- Testing. Man skal arbejde testdrevet. Altså for hver feature skal du skrive testkoden først.
- Refactoring. Som udvikler skal du hele tiden overveje, hvordan du kan forbedre strukturen.

- 40-Hour Week. Ingen lange arbejdsuger, det vil sige 37 timer i Danmark.
- On-site Customer. Tæt kontakt til kunden, måske in-house.
- Coding Standards, der skal være fælles standard for kodekvaliteten.
- Collective Code Ownership, alle i teamet ejer sourcekoden og kan vedligeholde den.
- Pair Programming, hvor man sidder parvis og programmerer.
- The Planning Game, hvor hele teamet er med til at tidsestimere.

Betragter man ovenstående, kan man se at arbejdet i teams er afgørende, og at samarbejde sikrer kvalitet i udviklingsarbejdet.



Arbejde agilt som enkelt person med Personal Extreme Programming (PxP).

Men der findes projekter, hvor der ikke er et hold af udviklere. Omstændigheder gør at det er en enkelt person, som står for hele udviklingen. Her er det ikke muligt med par-programmering og andre tiltag, som erstatter tidligere metoders krav til dokumentation.

Til disse enkeltmandsprojekter er der fremkommet nogle tanker, hvor XP er blevet ændret, således at metoden støtter små udviklingsprojekter. Her er, de principper som kræver et team, blevet ændret således at en enkelt person stadig kan drage fordel af at anvende Extreme Programming.

De ændrede principper er:

The Planning Game.

I et udviklingsteam skal man sammen med kunden spille et "planlægnings-spil" hvor man får skrevet kravene til systemet ned på nogle kort, kaldet "user stories". Man vil så i fællesskab diskutere, hvorledes disse krav bør prioriteres. Udviklerne ved noget om systemarkitektur, mens kunden bedst kender sine egne ønsker til systemet. Som eneudvikler bør du også sammen med kunden få lavet "user stories" og du skal også diskutere prioriteringen direkte med kunden. Selv hvis der ikke er nogen decideret kunde til systemet vil det være en god idé at skrive "user stories" så kravene til systemet brydes ned i overskuelige dele. Et "Kanban board" er en god hjælp til at holde styr på kortene undervejs. En udfordring ved at være eneste udvikler er dog at du ikke kan "sparre med" de andre udviklere omkring prioriteringen - du skal altså selv tage ansvaret for hvad der bør laves først set fra et udviklersynspunkt.

Pair Programming.

Denne praksis går ud på at udviklere to og to og sidder ved samme maskine og skriver kode. Dette er selvsagt umuligt at gøre som eneudvikler. Men du kan i stedet vælge:

- Enten at styrke "test først"-princippet ved at skrive meget grundige test så flere hjørner prøves af og kodekvaliteten derved øges.
- Eller du kan undervejs i kodelærbedet lave uformelle gennemgange af koden, hvor du

(højt!) fortæller dig selv, hvad koden rent faktisk gør. På den måde tvinger du dig selv til at kigge på koden fra en lidt anden synsvinkel. Og måske kan du således selv opdage problemer som din makker i par-programmering ellers normalt ville få øje på.

#### Collective Code Ownership.

I et team betyder dette princip at alle har et fælles ansvar for hele kodebasen og dermed også både ret og pligt til at ændre i den kode andre udviklere laver. Som eneudvikler er det din opgave at sørge for at dokumentere dit arbejde så grundigt at andre efterfølgende kan vedligeholde koden. Du vil ikke få fordel af at få andre udviklers mening. Du er den eneste som kender ideerne bag de fundne løsninger; og det er vigtigt at bevare den viden så den ikke bare forsvinder i det øjeblik projektet er slut for dig.

#### Coding Standards:

I et team skal alle følge samme kodestandard, således at alle udviklere let kan læse og forstå hinandens kode. Som eneudvikler skulle man umiddelbart tro at dette er let, men du skal også huske at hver gang du for eksempel henter kodestumper fra nettet bør du altid sørge for at standarden svarer til din egen. Du skal til stadighed huske på, at selv om du jo arbejder alene, skal dine tanker kunne forstås af andre som måske senere får til opgave at vedligeholde din kode. Man kan eksempelvis også forklare sin kode via WikiForms eller skrive readme-filer ved hjælp af markdown-sproget. Markdown er et letvægtssprog til ren tekst med en simpel syntaks, som også kan konverteres til html.

#### [Kom i gang.](#)

Arbejder du selv i projekt, hvor du er ene udvikler, så er PxP absolut en metode du bør

overveje. Er du træt af bureaukratiske arbejds gange eller projekter, som sejler tag et kig på PxP, som hverken er "for tung" eller "for let".

Denne artikel er inspireret af en nogle tanker fremsat af Ravikant Agarwal og Davis Umphress fra Auburn Universitet i Alabama.

*Sønderborg den 28.november 2018.*

*Tommy Haugaard*

*Karsten Skov*

*Adjunkt*

*Lektor*