

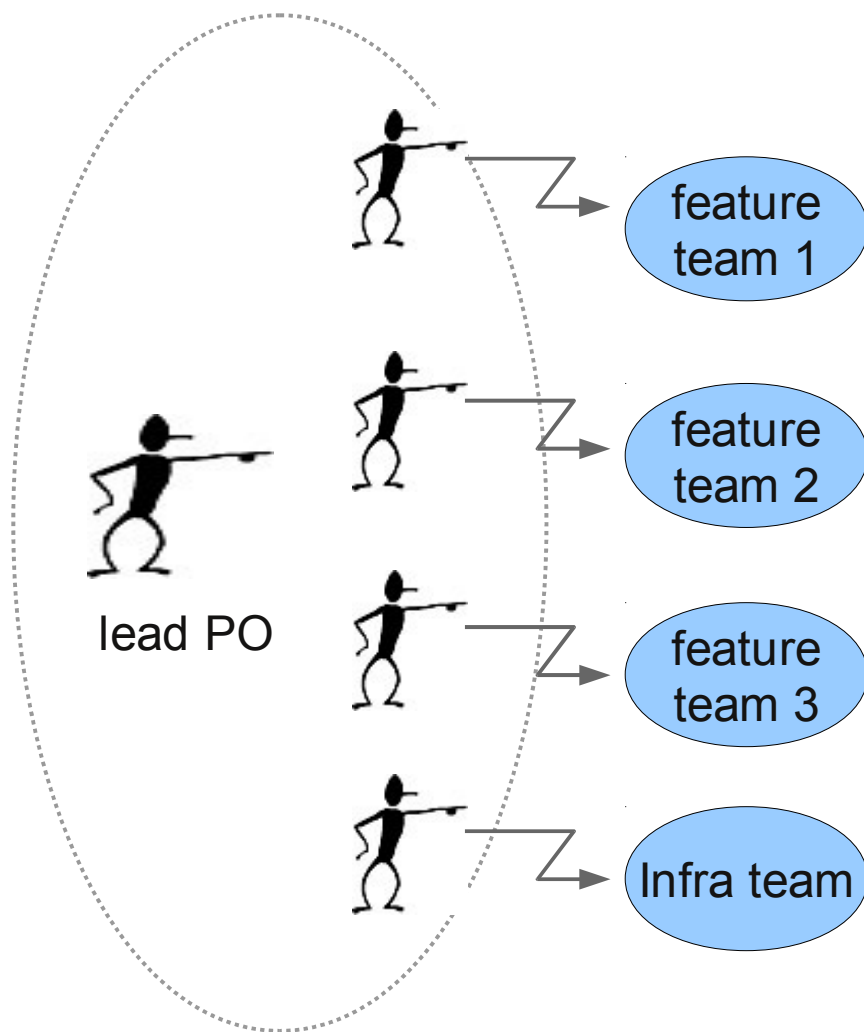
## Consultants United, april 2010, De agile projectorganisatie – de Product Owner

Om het onderwerp van dit artikel in te leiden wil ik eerst even terug naar de basis: Wat maakt een project agile? Ik denk dat een project agile is als het kan inspelen op veranderende omstandigheden. Wat voor projectorganisatie heb je hiervoor nodig? We weten dat een agile project via regelmatige afstemming met de klant de geleverde functionaliteit optimaliseert binnen de project-timebox. Optimaliseren betekent hier: doorlopend maximaliseren van de waarde voor de business van de lijst van nog te implementeren product features. Het mag duidelijk zijn dat voor een agile project een goede aansluiting op de klant (de “business”) dus cruciaal is. In mijn ervaring ligt juist hier een van de moeilijkste aspecten van vooral een groter agile project. Een aspect dat onderbelicht blijft in de agile literatuur van de afgelopen jaren. Hoe sluit je op een agile manier de business aan op de IT?

Laten we starten met het -altijd in de literatuur gebruikte- eenvoudige voorbeeld van een enkel Scrum team met een Product Owner en een Scrum Master. Het doel is om een IT-product te bouwen om de business-strategie van de organisatie te ondersteunen. De Scrum Master zorgt ervoor dat het Scrum proces “loopt”; deze rol bespreken we hier verder niet. De Product Owner (PO) heeft vanuit de business het mandaat om ervoor te zorgen dat het ontwikkelteam de goede dingen bouwt. Om dit te bereiken heeft de PO het druk met:

- samen met de “business” in het bedrijf globaal definiëren van het te bouwen product
- opstellen van een lijst van te implementeren features (bijvoorbeeld in de vorm van requirements, use cases of user stories)
- schatten van de complexiteit van elke feature
- plannen van de features in sprints (time boxes)
- per sprint het ontwikkelteam een opdracht geven en het resultaat accepteren, en
- continu tussentijdse veranderingen of verfijningen van het product verwerken in de lijst van nog te bouwen features.

Op deze manier vormt de PO dan de brug tussen business en IT en speelt hij een belangrijke rol in de agile organisatie. Bestond deze rol al in uw organisatie voordat u met Agile aan de slag ging? Ik heb het tot nu toe nog niet meegemaakt. Door de combinatie van business-oriëntatie, planning en team-aansturing is de PO ook nogal een “schaap met vijf poten”. Om het praktisch te houden bij de invoering van agile werken bekijken we welke bestaande rol of functie het beste bij de PO-rol past. De overige eigenschappen van de “ideale” PO vullen we dan met wat kunst-en-vliegwerk in. Nog een ander punt over “business agility”: hoe vul je de PO rol in bij een groot project met meerdere teams? En hoe sluit je dit aan op de rest van de bestaande organisatie? Het volgende plaatje geeft een voorbeeld uit mijn eigen praktijk.



Een project met vier Scrum teams bouwt een applicatie voor medische beeldbewerking. Een van de teams (in India) is verantwoordelijk voor de technische infrastructuur waarop de andere drie teams (in Nederland) features bouwen. Door de projectgrootte is de “organisatorische afstand” tussen business en implementatie-team nogal groot. Van de bestaande product manager mogen we natuurlijk niet verwachten dat die direct een of meerdere technische teams als PO aanstuurt. Daarom benoemen we per team een (gelukkig) al bestaande technisch product specialist tot PO. Zo'n specialist had al als taak om features te definiëren van een deelproduct, samen met product management en gebruikers (zoals medisch specialisten). Het stuk van de PO-rol dat we moeten toevoegen is het opstellen en bijhouden van de product backlog van een team. Elk van de vier PO's houdt voor zijn team een planning bij die vermeldt welke features in welke sprint worden opgeleverd. De PO's onderling stemmen de afhankelijkheden af en komen zo tot een masterplanning van het project. Deze planning is het eigendom van de lead PO. Vanwege de sterk technische aard van het project is gekozen voor de project architect als lead PO. Die stemt af met de product manager, die met de project leider en de architect in het “core team” zit.

Zijn we er hiermee? Er bleken wat ongewenste effecten op te treden. De PO's gingen ineens veel tijd besteden aan planning, iets wat niet bij hun functie en carrièrepad past. Wij hebben de PO's ontlast door een van de Scrum masters te laten assisteren bij de daadwerkelijke planning. Maar: elke PO blijft wel verantwoordelijk voor de deliverables van zijn team.

Bij dit voorbeeld komen we nog een belangrijk aspect van Agile tegen: hoe je omgaat met onderlinge afhankelijkheden tussen de leveringen van de teams. De klassieke 'oplossing' is om bij de projectstart een gedetailleerd overzicht van van afhankelijkheden te maken. En na enkele weken ontwikkeling klopt de planning al niet meer.. Wij hebben aan het begin van het project slechts een globale projectplanning gemaakt, waarin de meest voor de hand liggende afhankelijkheden staan.

Hiernaast wordt er van week tot week een gedetailleerde integratieplanning gemaakt. Het project bleek zo complex dat deze planning niet meer dan twee weken vooruit gemaakt kon worden! We hebben een dagelijkse standup meeting georganiseerd tussen POs, Scrum masters en testgroep over hoe de integratieplanning voor de komende dagen eruit zag. Met andere woorden: de detailplanning wordt van dag tot dag direct door de betrokkenen bijgewerkt.

Is de projectorganisatie daarmee nu echt agile? Ik denk dat we aardig op weg zijn in dit voorbeeld. Maar: hoe groter hoe minder agile. Hoe graag we ook willen, een groot complex project is minder flexibel dan een klein team met enkele developers en een Product Owner. Verder is het voor de business organisatie een hele cultuuromslag: in plaats van bij de projectstart een “contract” af te spreken met de ontwikkelgroep is de business nu via de PO in de lead tijdens de projectuitvoering. En moet de business al tijdens de projectgang in nauw overleg met het project reageren op veranderingen (zoals tegenvallers...).

En er zijn natuurlijk nog meer relevante aspecten, zoals agile test en acceptatie. Maar hierover een volgende keer meer!