

Wat is de rol van een projectleider in een Extreme Programming project?

Door Willem van den Ende. 11 Augustus 2003

Dat laat zich het best met een woord samenvatten: *faciliteren*. Dat betekent, dat de projectleider de voorwaarden schept voor een goedlopend ontwikkelteam en klantteam. Ook zorgt hij ervoor, dat eventueel betrokken hogere managementlagen op eenvoudige manier inzicht hebben in de voortgang van het project, zodat management en klantteam samen kunnen beslissen wanneer ze willen stoppen met software-ontwikkelen.

Dit klinkt misschien eenvoudig. Zorg ervoor dat de klanten en programmeurs met elkaar in gesprek raken en een overzichtelijk proces volgen. Voordat de situatie in bovenstaande alinea werkelijkheid kan worden is er wel het een en ander te doen. Daarna bewaakt de projectleider het project, zodat de communicatie soepel blijft verlopen en het proces in stand blijft.

In dit artikel wil ik, op basis van mijn ervaring in een aantal XP projecten, een paar facetten laten zien van wat een projectleider voor een xp project kan betekenen. Tevens wil ik een paar belangrijke valkuilen laten zien.

Communicatie tussen klanten en ontwikkelaars

Klanten geven aan wat ze willen, en welke functionaliteit ze het belangrijkste vinden, programmeurs schatten in hoeveel werk dat ongeveer is, en gaandeweg onstaat er zo een geprioriteerde planning. Het belangrijkste wat de projectleider hier moet doen, is ervoor zorgen dat de klanten en de programmeurs zich vrij voelen voor hun eigen belang op te komen, zodat er gaandeweg een behoorlijke verzameling goed ingeschatte stories is, op basis waarvan de klanten een aantal weken vooruit kunnen kijken.

Hoe faciliteert een projectleider de communicatie tussen de deelnemers aan het project? Allereerst draagt de projectleider zorg voor een goede werkomgeving van programmeurs en, als ze in hetzelfde pand werken, klanten. Communicatie binnen het project verloopt het eenvoudigst als iedereen die bij het project betrokken is, in één ruimte zit. De projectleider zorgt ervoor dat die ontwikkelruimte er komt, wat in de meeste organisaties vrij bewerkelijk is. Daarna kost het meestal ook nog wat moeite om alle betrokkenen, met name klanten, ervan te overtuigen dat ze ook dagelijks in de ontwikkelruimte aanwezig moeten zijn. Als het niet lukt om iedereen intern te verhuizen, dan ligt er voor de projectleider werk om er op te letten dat programmeurs en klanten elkaar vaak genoeg spreken en zien. Dit kan bij voorbeeld door elke ochtend een standup-meeting houden, en klanten en programmeurs te activeren om zelf standup-meetings te organiseren gedurende de dag als er een probleem ligt dat gezamenlijk opgepakt dient te worden.

Hoe kan een projectleider zien of er effectief gecommuniceerd wordt? De periodieke planning game is hier meestal een goede graadmeter voor. Een goed lopende planning game bestaat uit drie onderdelen:

1. De klant accepteert (of weigert) gerealiseerde functionaliteit formeel. De ontwikkelsnelheid wordt berekend op basis van geaccepteerde functionaliteit.
2. Klanten en ontwikkelaars bespreken nieuwe functionaliteit; de ontwikkelaars schatten de kosten van elke story in.
3. De klant bepaalt welke functionaliteit er het eerst door de programmeurs gerealiseerd moet worden.

Als de planning game niet goed loopt is dat vaak te zien aan het door elkaar heen lopen van deel een en twee. Als de klanten niet dagelijks het werk hebben bekeken, en om wijzigingen hebben gevraagd, moeten ze in een keer alle nieuwe functionaliteit begrijpen en beoordelen. Voor functionaliteit die niet helemaal naar wens is gemaakt kunnen ze eigenlijk alleen zeggen 'dit is niet wat ik wil' en dan niet accepteren, wat goodwill van de ontwikkelaars kost, of weer nieuwe stories schrijven, wat tijd en energie kost.

Het eerste deel van de planning game kan alleen soepel verlopen als er gedurende een iteratie effectief is gecommuniceerd tussen klanten en ontwikkelaars. Informele acceptatie tijdens de iteratie is dan tot stand gekomen doordat de klant dagelijks verse functionaliteit heeft bekeken, en van commentaar heeft voorzien, waarna de ontwikkelaars kleine wijzigingen aanbrengen op basis van het voortschrijdende inzicht van de klant. Aan het eind van de iteratie is dan alles naar wens gerealiseerd, waardoor formele acceptatie weinig tijd meer kost.

De projectleider zorgt er dus voor, dat de klant vaak genoeg het systeem bekijkt, en vragen stelt, en dat de ontwikkelaars de klant op de hoogte houden van de vorderingen. Daarnaast is het belangrijk om in de gaten te houden of de klanten en ontwikkelaars nog voldoende vertrouwen in elkaar hebben. Ook in een Extreme Programming project zit de wind wel eens tegen, al wordt er nog zo goed getest, gerefactored enzovoorts. Als het vertrouwen afneemt, wordt het lastiger om effectief tussen klanten en ontwikkelaars te overleggen.

De grootste val waar een projectleider in dit geval in kan stappen is om de verantwoordelijkheid voor communicatie van klanten en ontwikkelaars over te nemen. De projectleider treed nu op als intermediair, en kan klem komen te zitten. Een intermediair zorgt voor extra indirectie, en dus mogelijke misverstanden. Zowel klanten als programmeurs proberen hun verantwoordelijkheid op de projectleider af te schuiven. De projectleider heeft helemaal niet zo veel middelen in handen om klanten of programmeurs van gedrag te laten veranderen zodat hij zijn doel kan bereiken.

Toch is het o zo makkelijk om in deze val te stappen. Klanten moeten in een Extreme Programming of Agile project meestal meer werk verzetten dan ze gewend zijn. Vaak laten klanten het bedenken van functionaliteit het liefst over aan software ontwikkelaars of projectleiders, en klagen achteraf pas, dat het 'niet is wat ze willen'. Nu worden klanten zelf verantwoordelijk gemaakt voor het formuleren van functionaliteit. Ze komen er achter, dat ze helemaal niet zo precies weten wat ze willen als ze wel dachten. Het is niet uitzonderlijk dat klanten in het begin moeite hebben om zelfs maar voor een week functionaliteit te bedenken. Ook vinden ze vaak dat ze wel erg veel tijd aan het overleg met de programmeurs moeten besteden. Hier zal de projectleider, samen met de XP coach, stevig in zijn schoenen moeten staan en volhouden dat de klant zelf stories blijft bedenken. Na verloop van tijd gaan de klanten namelijk merken, dat hun extra inspanning iets bijzonders oplevert: *ze krijgen wat ze willen*. Tijdens onze projecten peilen we regelmatig de stemming in het project, onder andere via mini-retrospectieven. Het blijkt dan dat klanten op het moment dat ze beseffen dat ze kunnen krijgen wat ze willen, ze tegelijkertijd ontdekken dat ze veel tijd in het project moeten steken. Het kost blijkbaar wat tijd om het verband tussen *krijgen wat je wilt* en *eigen tijd investeren* te zien.

Uiteindelijk kan deze inspanning van de klant de totale project-inspanning (programmeurs, klanten, projectleider etc.) verkleinen, omdat er doelmatiger aan gewenste functionaliteit wordt gewerkt. Als de klant snel zegt (snelle feedback) wat hij wel en niet wil, wordt er door de programmeurs minder tijd besteed aan ongewenste functionaliteit en aan het raden van klantwensen. Het is voor een programmeur ook minder demotiverend om na een uur werk te horen *kan dat niet*

anders? dan dat je dat na een week, maand of zelfs een jaar hoort. Bij zulke snelle feedback hebben programmeurs de kans snel aanpassingen te maken. Aan het eind van een iteratie spreekt de klant zijn waardering uit en zegt: *dit is precies wat ik wil*. Voor zowel klanten als ontwikkelaars is dit een prettige manier om te werken.

Is het eten nog wel net zo lekker, als je weet hoe de kok werkt?

Er kleeft nog wel een risico aan intensieve communicatie tussen klanten en ontwikkelaars gedurende een iteratie. Een klant kijkt als het ware mee in de keuken met de ontwikkelaars, en dat is niet altijd een fraai gezicht. Een klant zal soms over een sterke maag moeten beschikken om het eten te verteren.

We hadden het eerder al over het vertrouwen dat een klantteam in een ontwikkelteam moet hebben om effectief samen te kunnen werken. Een voorbeeld van zo'n situatie is als het ontwikkelteam onverwacht een defect vindt. In plaats van dat degene die het defect vindt het stilletjes in zijn eentje gaat 'fixen', zodat de klant er niets van merkt, legt hij het werk stil, en vraagt zijn collega's bij het whiteboard te komen voor een extra stand-up meeting. De programmeur doet dit, om iedereen in het team op de hoogte te stellen van het defect, het defect grondig te repareren zonder dat er tijdens het repareren nieuwe defecten ontstaan, en de oorzaak van het defect te achterhalen, zodat dit soort defecten de volgende keer voorkomen kunnen worden.

De klant zou gemakkelijk kunnen denken: “zijn ze nu *alweer* aan het vergaderen? Ik heb liever dat ze aan het werk gaan. Ze zitten ook al de hele tijd in groepjes van twee te kletsen¹.” Als je niet, zoals veel programmeurs, uit eerste hand weet hoeveel tijd verloren kan gaan met het 'fixen' van een defect kan dat gemakkelijk zo lijken. Die tijd, desnoods tot diep in de nacht, kan gaan zitten in het in je eentje speuren naar het defect, het aanpassen van het programma, het weer terugkeren van hetzelfde defect daarna omdat de andere teamleden de reparatie niet begrijpen, of het oplossen van defecten die tijdens de 'reparatie' gemaakt zijn.

Het is het goed recht van de klant om zich te verbazen over de energie die zichtbaar wordt gestoken in het verhelpen van een defect, hij of zij mag ook vragen waarom deze extra vergadering wordt gehouden. Het blijft het recht van de programmeurs om te bepalen hoe ze een defect verhelpen, omdat ze het recht hebben om kwalitatief goed werk op te leveren. De projectleider moet er voor zorgen dat dit recht gewaarborgd blijft, door ervoor te zorgen dat programmeurs goed in staat zijn uit te leggen waarom ze hun werk op een bepaalde manier doen, en, desnoods als de klant zich er teveel mee blijft bemoeien, als scheidsrechter op te treden en de klant terug te fluiten.

Te veel bemoeienis met de werkwijze van het ontwikkelteam is demotiverend voor de programmeurs. Klanten en programmeurs horen te overleggen over functionaliteit en planning. Het is niet aan de klant om te bepalen of een defect gerepareerd wordt door middel van bij voorbeeld extra unittests, debuggen of een stand-up. Als klanten voortdurend tijdens planning games en stand-ups over het proces willen praten, of als ze buiten de planning games om aan individuele programmeurs vragen om 'dit effe te fixen' moet de projectleider ingrijpen en de werkwijze apart met het klantteam te bespreken. Het is natuurlijk beter als de ontwikkelaars zelf hun werkwijze kunnen verdedigen en deze vasthouden. Als een klant op een programmeur afkomt met een verzoek om 'dit effe te fixen' kan hij of zij er natuurlijk voor kiezen een extra stand-up te houden, of naar de planning game te verwijzen.

1 De programmeurs vinden dat ze aan het pairprogrammeren zijn..

Het omgekeerde kan natuurlijk ook gebeuren, dat het ontwikkelteam zich gaat bemoeien met wat de klant wil. Zo lang het klantteam het onderling eens is over wat ze willen, mogen ze zo vaak van gedachten veranderen als ze willen. Dat is voor programmeurs wel eens vervelend: de klant klanten weten soms niet weet wat ze willen, en gebruiken de ervaring die ze op te doen tijdens de bouw van het systeem om te leren, en dat is hun goed recht. Het is dan voor programmeurs wel erg gemakkelijk om ongevraagd advies te geven over functionaliteit of te zeggen dat een wijziging niet kan. Als het goed loopt, zeggen de programmeurs alleen wat het kost om van gedachten te veranderen, en komen de klanten voor recht op om te bepalen wat functionaliteit is, en wat niet. Zo niet, dan is de projectleider er ook voor om de programmeurs terug te fluiten.

Ontwikkeling van het ontwikkelteam

Het belangrijkste 'product' wat een projectleider levert is een grotendeels zelfstandig functionerend ontwikkelteam. Twee belangrijke voorwaarden om zo'n team te ontwikkelen zijn opleiding en werkomstandigheden.

Opleiding

Het belangrijkste wat de projectleider aan opleiding kan doen is de samenstelling van het team in de gaten houden, zowel aan het begin als tijdens het project. Bij voorkeur bestaat er in een ontwikkelteam een gezonde verhouding tussen het aantal ervaren en minder ervaren programmeurs. Het is niet per se noodzakelijk om aan het begin van een project iedereen in de gebruikte technologie op te leiden. Als er een paar mensen zijn die er mee overweg kunnen, kunnen ze door pairprogrammeren hun kennis aan de anderen overbrengen. Varieteit aan kennis en ervaring werkt tijdens de loop van een project vaak positief, omdat er zelden een heel project gedaan kan worden met precies die technologie die in het begin is uitgezocht. Belangrijker dan keuze van technologie is in Extreme Programming ervaring met de werkwijzen ('practices'). Als geen van de deelnemers ervaring heeft met bijvoorbeeld test-driven development of refactoring, is het raadzaam daar extra tijd voor in te plannen en/of een XP coach in te schakelen. Als een projectleider voldoende tijd en vaardigheden heeft om werk binnen het ontwikkelteam te doen, kan hij eventueel ook de rol van XP Coach vervullen.

Werkomstandigheden

De projectleider zal er waarschijnlijk ook op moeten toezien dat de ontwikkelaars voldoende krachtige computers, leesbare schermen en goede bureaus en stoelen hebben. Slecht gereedschap is een demotiverende factor voor de ontwikkelaars. In verhouding tot de loonkosten van een project kosten computers en dergelijke relatief weinig geld.

Voor het overige kunnen programmeurs zelf veel invloed uitoefenen op de werkomstandigheden, bij voorbeeld door het maken van automatische builds, het gebruik maken van versiebeheer, en het uitkiezen van handige software ter ondersteuning van het programmeren. Gelukkig is hier voor tegenwoordig veel open source software beschikbaar, zodat programmeurs hiervoor niet om budget hoeven te vragen.

Planning en Tracking

Als de planning games goed lopen, is het eenvoudig naar buiten een overzichtelijke planning te presenteren. Een belangrijk aandachtspunt is bemoeienis met de planning door hoger management; de projectleider zorgt ook bij planning voor

een context waarin het team kan functioneren.

De projectleider kan dit doen, door te zorgen dat zowel taken als management-ingrepen gebufferd worden, zodat het team niet continu over veranderingen hoeft na te denken, maar er periodiek gestructureerd mee bezig kan zijn, bij voorbeeld tijdens de planning game of een standup.

De takenbuffer houdt in, dat al het werk waar het team (nog) niet mee bezig is extern mag worden hergeprioriteerd. Story kaartjes die in de plannig game zijn ingediend, maar waar nog geen werk voor verricht is, mogen nog binnen een iteratie worden vervangen door stories die evenveel werk zijn. Tevens mogen geplande stories binnen een iteratie door het klantteam van volgorde worden gewijzigd. Het wijzigen van volgorde kan relevant zijn als blijkt dat de ontwikkelsnelheid in een iteratie ineens lager is, of de klant met een verandering van stories een strategische doelstelling beter kan dienen.

Als management-ingrepen buffer schermt de projectleider het team afs voor organisatorische wijzigingen die in hogere managementlagen worden bedacht. De projectleider geeft alleen de echt belangrijke wijzingen door, dus bij voorbeeld geen wijzigingen die nog bediscussieerd worden. Kleine managementsignalen worden soms door het ontwikkelteam versterkt of kunnen niet goed in context geplaatst worden en leiden dan tot verstoring.

Conclusie

Een projectleider doet meer dan ik in de afgelopen paar pagina's heb kunnen schetsen. Ik hoop u hiermee een beeld te hebben gegeven van een paar aspecten die een XP project kunnen maken of breken. Met name de interactie tussen het klantteam en het ontwikkelteam kunnen een projectleider veel hoofdbreken bezorgen. Deze hoofdbreken kunnen met wat creativiteit en doorzettingsvermogen zorgen voor het voldane gevoel van soepel lopende planning games, en een project dat ruim binnen de tijd een naar tevredenheid werkend product oplevert.

Dit artikel is mede tot stand gekomen door opmerkingen van een aantal enthousiaste reviewers, die ik bij deze wil bedanken: Pascal van Cauwenberghe, Marc Evers, Nynke Fokma, Erik Groeneveld, Marko van der Puil en Peter Schrier. Uw commentaar is welkom voor een eventuele volgende versie van dit artikel.

Willem van den Ende is werkzaam als Software Development Programming Coach. Hij begeleidt diverse bedrijven bij het introduceren van Extreme Programming en Agile werkwijzen. Voor vragen en opmerkingen naar aanleiding van dit artikel is hij te bereiken onder willem@willemvandenende.com