Jeroen Boydens, Marko van Dooren en Eric Steegmans zijn allen leden van de onderzoeksgroep softwareontwikkelingsmethodologieën, Katholieke Universiteit Leuven

XP: eXtreme Programming

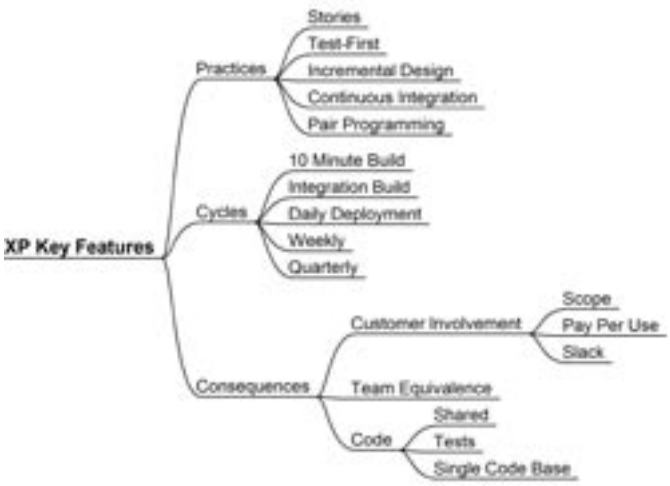
HET SOFTWAREONTWIKKELPROCES XP IN .NET-PROJECTEN

Dit artikel bevat een omschrijving van de essentiële onderdelen van het ontwikkelproces eXtreme Programming (XP). Eerst zetten we een aantal gebeurtenissen en gevolgen in het XP-proces op een rij, waarna we een SWOT-analyse (sterkte, zwakte, mogelijkheden, bedreigingen) geven om de succesfactoren van het XP-proces te identificeren. We besluiten met een overzicht van de belangrijkste voordelen en aandachtspunten van het XP-proces.

Het XP-ontwikkelproces bestaat uit een aantal 'practices' - zie afbeelding 1. We lichten de volgende onderdelen nader toe: de stories, de testen, het incrementeel ontwerp, de continue integratie en het programmeren per paar.

Practices binnen het XP-proces: stories

Een eerste onderdeel zijn de verhalen, de 'stories' die aan de basis liggen voor het ontwikkelen van nieuwe softwareonderdelen. Uit een dialoog met de eindgebruikers zoekt de ontwikkelaar naar de verschillende functionaliteiten die het eindproduct moet bevatten. Voor dit zoekproces gebruikt hij geen specifieke methodiek. Dit in tegenstelling tot andere ontwikkelprocessen. De eerste stories die de ontwikkelaar uit de gesprekken afleidt, dienen voor de opstart van het totale XP-proces. Hij zal deze stories in de loop van het proces nog geregeld uitbreiden en wijzigen. Het is immers zo dat hij op elk moment in het proces alleen die functionaliteit implementeert waar zowel de klant als hijzelf het maximale belang aan hechten. Gedurende het volledige proces duidt de klant aan welke stories essentieel zijn in een eindproduct en welke stories van secundair belang zijn. Zo kan de ontwikkelaar een prioriteitenvolgorde vastleggen in de stories en kan hij zich continu concentreren op de onderdelen die de meeste toegevoegde waarde aan het eindproduct aanbrengen. Deze volgorde zal de ontwikkelaar telkens bijstellen wanneer hij nieuwe stories toevoegt. Door de onderhandelingen over de stories met de klant vermijdt de ontwikkelaar het risico op het afleveren van een product dat niet voldoet aan de verwachtingen van de klant.



Afbeelding 1. De belangrijkste kenmerken van XP

Testen

Een tweede onderdeel van 'practices' binnen het XP-proces zijn de testen. Net als bij andere ontwikkelprocessen zal de ontwikkelaar binnen XP maximale aandacht besteden aan het testen van de software. Bij het implementeren van een story maakt hij daarom eerst de nodige testen om de correctheid van de ontwikkelde code te verifiëren. Deze testen omvatten zowel de eenheidstesten (unit-tests) als de functionele testen en de integratietesten. Met de eenheidstesten verifieert hij het correct functioneren van één enkele routine. Hij schrijft deze eenheidstesten nog vóór hij de routine implementeert. Met de functionele testen controleert hij het overkoepelende geheel. Hij schrijft deze samen met de klant. Aan de hand van de integratietesten ziet hij erop toe dat door het toevoegen van nieuwe onderdelen alle bestaande onderdelen blijven draaien. Ook deze integratietesten bouwt hij in overleg met de klant, die het probleemdomein grondig kent.

Incrementeel ontwerp

Het belangrijkste onderdeel van de 'practices' binnen XP is het incrementeel ontwerp. De ontwikkelaar voegt telkens kleine onderdelen toe aan het product in de vorm van stories. Zo bouwt hij stapsgewijs de totale oplossing. Bij het implementeren merkt hij dat bepaalde stories leiden tot nieuwe stories. Deze laatste voegt hij toe aan de stapel later uit te voeren stories. Hij is pas klaar met de implementatie van een story, wanneer alle testen voor de story correct verlopen. Hij is zowel verantwoordelijk voor het correct functioneren van de code die hij net geschreven heeft als voor de integratie van deze code in het geheel. Wanneer de integratietesten falen, kan dit ertoe leiden dat hij andere stories moet aanpassen en opnieuw moet testen. De verschillende cycli, die staan vermeld in afbeelding 1 komen hier duidelijk aan bod. De ontwikkelaar voert op regelmatige tijdstippen het buildproces uit (10 minute build). Tijdens dit proces controleert hij automatisch de testen voor dit specifieke deel. Wanneer hij klaar is met dit deel, voert hij de integratie-build uit. Dit is een volgend onderdeel van de 'practices' binnen XP, namelijk de *continue integratie*. Deze continue integratie leidt tot een werkend product met nieuwe functionaliteit. In principe kan men dagelijks een deploymentversie maken om ter beschikking te stellen aan de eindgebruiker, zodat deze de software al kan gebruiken in het stadium waar de ontwikkeling nu gekomen is. Een ruimere cyclus is de weekcyclus waar de ontwikkelaar en de eindgebruiker samen beslissen welke stories er die week topprioriteit hebben. Ten slotte kan de kwartaalcyclus een aantal mijlpalen binnen het softwareproces in kaart brengen.

Programmeren per paar

Een laatste onderdeel van 'practices' is het programmeren per paar: alle ontwikkelwerk wordt telkens door twee personen samen ver-

richt. Het paar neemt de verantwoordelijkheid om een gekozen story volledig te implementeren. Door telkens per paar te werken verspreidt de kennis van allerhande strategieën zich snel in de ontwikkelgroep. De samenstelling van de paren verandert voortdurend bij de implementatie van nieuwe stories. Op die manier kan elke ontwikkelaar alle code van het totale product van dichtbij zien, want tijdens de integratiefase moet hij soms onderdelen van anderen wijzigen. Dit vereist dat de ontwikkelaars een aantal afspraken maken omtrent coderingsstijl en documentatiestijl.

Gevolgen van het XP-proces: consequences

Het XP-proces heeft een aantal zeer directe gevolgen, zie 'Consequences' in afbeelding 1. Een eerste gevolg is dat de klant continu samenwerkt met het ontwikkelteam. Hij formuleert zijn initiële verwachtingen van het project. Hij stuurt, in voortdurend overleg met de ontwikkelaar, de prioriteiten van de stories bij. Bovendien kan hij tijdens het proces beslissen om bepaalde stories niet verder te laten uitwerken, zonder dat dit gevolgen heeft voor het reeds geleverde werk van de ontwikkelaars. Ten slotte kan hij vereisten, die aanvankelijk vergeten zijn, op latere tijdstippen toevoegen. Deze nieuwe vereisten leiden tot bijkomende stories die ingeschoven worden in de prioriteitenlijst. Hij betaalt enkel de ontwikkeling voor die stories die binnen het afgesproken tijdschema vallen. Hierdoor wordt tevens het risico op falen van dit project verlaagd. Om de klant een idee te geven van een mogelijk eindproduct kan het team een aantal stories extra opsommen in een aanvangsfase ('slack'). Later kan een aantal van die extra stories weer wegvallen, wanneer de klant een duidelijk beeld krijgt van de functionaliteit die hij in het product wil. Tengevolge van de continue integratie kan de klant ook regelmatig een nieuwe partiële versie van het product in gebruik nemen.

Een ander gevolg van de XP-strategie is dat het ontwikkelingsteam horizontaal gestructureerd moet zijn. Elke ontwikkelaar neemt immers de verschillende rollen uit het klassieke ontwikkelproces op zich. Deze rollen zijn die van analist, architect, implementator, tester, enzovoort. Het idee van hiërarchie geldt hier niet: het team bevat geen leider of hoofdprogrammeur die het volledige proces stuurt. Alle teamleden nemen hun eigen verantwoordelijkheid in het proces op. Wel kan een coach de nodige sturing geven aan de groep en hen erop wijzen dat een bepaald niveau van refactoring van belang is voor de overkoepelende architectuur.

Een laatste gevolg van de XP-strategie is dat er geen code toegekend wordt aan één programmeur of aan een groep programmeurs. Alle ontwikkelaars kennen het codeerwerk van elkaar en accepteren wijzigingen in elkaars code. Dit levert een bijkomend voordeel dat wanneer een ontwikkelaar het team verlaat, de gevolgen voor het project beperkt blijven. Ook de testen op deze gewijzigde onderdelen kunnen zij uitbreiden of corrigeren. Er is zodoende voor dit XP-proces voorzien in een versiecontrolesysteem, zodat alle stabiele onderdelen voor de volledige groep toegankelijk zijn.

Sterktes van het XP-proces

Een aantal voordelen van het XP-proces is te zien in afbeelding 2 (XP Swot: Strengths). Een eerste is dat we bijna altijd een beschikbaar eindproduct hebben om mee te geven aan de klant. Door de continue integratie kan de klant de software in een zeer vroeg stadium gaan gebruiken. Zo kan de gebruiker bepalen welke uitbreidingen hij alsnog wil toevoegen. Bepaalde onderdelen die aanvankelijk waren vereist kan hij weglaten ten gunste van de implementatie van andere onderdelen. Wijzigingen in vereisten van het eindproduct, of niet specifieke beschrijving van dit eindproduct, leveren voor XP geen problemen op, omdat op geen enkel ogenblik in het XP-proces een design van het softwaresysteem vastgelegd wordt.
Doordat de vereisten van het product kunnen wijzigen, voldoet deze strategie ook aan de 20/80-regel. De eerste 20% van het uitgevoerde werk door de ontwikkelaar, namelijk de 20% belangrijkste stories, omvatten 80% van de gebruikte functionaliteit bij de klant. Onderdelen van het eindproduct die de klant slechts zeer zelden zal gebruiken, zijn uitgesteld tot een later tijdstip in de ontwikkeling.

Afbeelding 2. XP SWOT-analyse

Testen en klantintegratie beïnvloeden het XP-proces ook positief. Het voortdurend testen leidt tot een correct functionerend eindproduct. De sterke integratie van de klant in dit proces resulteert in een eindproduct dat overeenstemt met de verwachtingen van de klant. We kunnen opmerken dat testen en klantintegratie ook voorkomen in andere ontwikkelprocessen, maar typerend voor het XP-proces is dat dit al het geval is in een zeer vroeg stadium.

Zwaktes van het XP-proces

Een aantal plaatsen waar XP voor wijziging vatbaar is, of richtingen die de ontwikkelaar beter kan ontzien bij het werken volgens XP, staan eveneens in afbeelding 2 (XP Swot: Weaknesses). We zijn ervan overtuigd dat grote projecten niet haalbaar zijn met een dergelijke strategie. In XP kijken we nooit verder vooruit dan de huidige cyclus. Hierdoor is het moeilijk om in een aanvangsfase de correcte architectuur te voorspellen. Door het gebrek aan deze initiële architectuur kan het zijn dat een groot deel van het uitgevoerde werk verloren gaat wanneer een essentiële architectuurwijziging zich voordoet.

Een andere zwakte van XP is het feit dat de resulterende code het enige communicatiemiddel is onder de ontwikkelaars, in tegenstelling tot andere ontwikkelprocessen waar naast de code ook UML-diagrammen of andere vormen van diagrammen deel uitmaken van het proces. Daardoor is het een onvoorwaardelijke vereiste dat in het XP-proces de code sterk gedocumenteerd moet zijn, wat zeer moeilijk af te dwingen is. Wegens het ontbreken van elke vorm van hiërarchie moeten we ook extra aandacht besteden aan het werken in teamverband. Het XP-proces moedigt het werken als gelijken aan, zodat alle leden van het team in staat zijn om elk onderdeel van het totale eindproduct te wijzigen.

Ten slotte is de wijze waarop de stories afgeleid worden niet vast omschreven. We starten met een initieel aantal stories, breiden ze dan verder uit tijdens de ontwikkeling en kunnen een aantal weer laten vervallen. Indien niet gestart wordt met die stories die een zware impact hebben op de uit te bouwen architectuur, kan dit zware gevolgen hebben voor een latere fase van het proces.

Mogelijkheden in het XP-proces

Een aantal mogelijkheden en interessante trends van het XP-proces staan in afbeelding 2 (XP Swot: Opportunities). Het XP-proces beschikt bijvoorbeeld over de belangrijke eigenschap om bijna continu een eindproduct ter beschikking te stellen. Het XP-proces is daarnaast uitermate geschikt voor een kleine ontwikkelgroep die bestaat uit ontwikkelaars van het zelfde niveau.

Ook door in de eerste stappen van het proces reeds gebruik te maken van vertrouwde architecturale patronen, kunnen heel wat problemen voorkomen worden. Patronen zijn immers structuren die reeds hun waarde hebben bewezen in andere softwareconfiguraties. De kans op fouten door ingrijpende architectuurwijzigingen wordt op deze manier gereduceerd. Verder steunen de huidige ontwikkelomgevingen en programmeertalen de ontwikkelaars in het uitgebreid invoeren van documentatie. Deze documentatie is essentieel voor het correct doorgeven van de informatie onder de verschillende ontwikkelaars. Als laatste voordeel van het werken in het XP-proces stellen we dat kleine verkennende projecten een nuttige basis vormen voor complexe of grote softwaresystemen.

Bedreigingen van het XP-proces

Als laatste onderdeel in de analyse rond het al dan niet slagen van een XP-proces moet expliciet gekeken worden naar bedreigingen die een dergelijk proces met zich mee brengt. Deze zijn te zien in afbeelding 2 (XP Swot: Threats). Al meermalen hebben we aangehaald dat deze strategie niet kan functioneren voor grote projecten. Maar ook dienen we aandacht te besteden aan projecten die groeien in omvang. Projecten die aanvankelijk geschikt leken om te ontwikkelen via XP kunnen zo groeien dat het moeilijk wordt met XP verder te werken. Wijzigingen in kleine projecten kunnen beperkt blijven. In grotere projecten kunnen vergelijkbare wijzigingen grote, onvoorzienbare gevolgen hebben. Een strategie zoals XP is in deze situatie echt uit den boze. Ook dienen we de nodige aandacht te besteden aan het feit dat bepaalde ontwikkelaars behoefte hebben aan expliciete sturing in een hiërarchie. Daarbij komt dat ontwikkelaars die gewend zijn om solo te programmeren, problemen kunnen ondervinden bij het coderen per paar.

Risico mislukt project verlaagd

We hebben aangetoond dat de ontwikkelaar over de nodige reserves moet beschikken: hij kan niet elk product volgens het XP-proces bouwen. We hebben de voorbeelden aangehaald van softwareproducten waar hij geen initieel idee kan vormen van de onderliggende structuur. We weten dat bij deze producten de structuur regelmatig wijzigt tijdens het XP-proces, met het gevolg dat codewijzigingen zich verspreiden in het systeem. Ook hebben we erop gewezen dat grote softwareprojecten niet geschikt zijn om ontwikkeld te worden via het XP-proces. Grote projecten zijn in aanvang zo omvangrijk dat een degelijke vereistenanalyse en architectuurstudie noodzakelijk zijn bij de opstart van het project.

XP in Visual Studio 2005

Wat biedt Visual Studio 2005 op het gebied van XP? Als eerste kunnen we Visual Studio Team System goed gebruiken om de stories op te slaan. Ook is het uitermate geschikt voor communicatie tussen de teamleden onderling en met de klant. Natuurlijk is er hulp van de geïntegreerde unit-tests en de andere tools. Maar er komt meer, Visual Studio Team System wordt standaard geleverd met twee programmeermethodieken: MSF (Microsoft Solution Framework) for Agile Software Development en MSF for CMMI (Capability Maturity Model Integration). Ontwikkelteams zijn niet beperkt tot deze twee MSF-methodieken. Visual Studio Team System is zodanig open opgezet dat andere ontwikkelmethodieken later toegevoegd kunnen worden. Verschillende leveranciers hebben al introducties gedaan. Microsoft zelf heeft aangekondigd ook ISO en Extreme Programming te gaan ondersteunen.

Tot besluit

Toch wijzen we erop dat de XP-strategie voor softwareontwikkeling een aantal voordelen biedt die in weinig andere ontwikkelingsprocessen terug te vinden is. De belangrijkste voordelen zijn de aanpasbaarheid bij wijzigingen in de vereisten van het product en het zogenaamd continu beschikbaar zijn van een afleverbaar product. Op zich zijn dit twee grote voordelen die in de softwarewereld vaak vereist zijn. We weten immers dat de vraag van de klant voortdurend wijzigt, omdat ofwel het probleemdomein niet voldoende uitgelegd is, of omdat nieuwe ideeën opduiken voor extra functionaliteit. Ook krijgt de klant het gevoel dat het product echt groeit naar zijn wensen en dat hij steeds over een product kan beschikken om mee proef te draaien.
Tevens hebben we aangetoond dat door gebruik te maken van het XP-proces het risico op het mislukken van het project aanzienlijk verlaagd is. De klant heeft continu inspraak op de stories in het proces, de ontwikkelaars implementeren de stories volgens prioriteit en de teamleden die vroegtijdig het project verlaten worden door hun partner opgevolgd.

Jeroen Boydens KHBO Dept IW&T, Jeroen.Boydens@khbo.be is verbonden aan een industriële hogeschool in België en is geaffilieerd onderzoeker bij de onderzoeksgroep softwareontwikkelingsmethodologieën aan de K.U.Leuven.

Marko van Dooren K.U.Leuven Dept CW, Marko.vanDooren@cs.kuleuven.ac.be is verbonden aan de K.U.Leuven in België en is onderzoeker bij de onderzoeksgroep softwareontwikkelingsmethodologieën aan de K.U.Leuven.

Eric Steegmans K.U.Leuven Dept CW, Eric.Steegmans@cs.kuleuven.ac.be is professor aan de K.U.Leuven in België en is coördinator van de onderzoeksgroep softwareontwikkelingsmethodologieën aan de K.U.Leuven. http://www.cs.kuleuven.ac.be

Nuttige internetadressen

http://www.extremeprogramming.org/
http://www.xprogramming.com/
"Extreme Programming Explained: Embrace Change, 2nd Edition" By Kent Beck, Cynthia Andres. Published by Addison Wesley Professional.
http://www.awprofessional.com/title/0321278658
"eXtreme .NET: Introducing eXtreme Programming Techniques to .NET Developers" By Neil Roodyn. Published by Addison Wesley Professional.
http://www.awprofessional.com/title/0321303636
http://www.theserverside.net/talks/library.tss#lorilamkin