

Ontwikkelen van mobiele applicaties

Verschillen en overeenkomsten met traditionele softwareontwikkeling

Jeroen Veldkamp MSc en Gerald Hemmers MA

Het aantal mobiele apparaten en het gebruik van mobiele applicaties is explosief gegroeid. Het is voor uw businesscase mogelijk interessant om een mobiele applicatie te ontwikkelen, omdat hoogstwaarschijnlijk een groeiend deel van uw doelgroep over een mobiel apparaat beschikt. Hierbij staat u voor een mobiele uitdaging, want het proces dat bij het ontwikkelen van mobiele software komt kijken, verschilt op een aantal punten van het ontwikkelproces van traditionele software. In dit artikel wordt een verkennende analyse gemaakt: op welke aspecten verschillen deze processen van elkaar, wat zijn de overeenkomsten en welke keuzes moeten er worden gemaakt?

Inleiding

Het gebruik van mobiele apparaten, zowel zakelijk als privé, beleeft een enorme groei. In het laatste kwartaal van 2011 is het aantal gebruikers van tablets verdubbeld ([IGFK11]). Met de enorme groei aan potentiële gebruikers neemt ook het aantal applicaties dat op deze apparaten gebruikt kan worden toe: eind 2011 waren er al bijna 420.000 applicaties beschikbaar op de Google Market en bijna 460.000 applicaties in Apple app store ([Horn11]). Naast deze applicaties die publiek beschikbaar zijn in de verschillende app stores, wordt er ook een onbekend aantal *in-house* applicaties ontwikkeld voor gebruik binnen bedrijven. Met dit enorme aantal applicaties dat wordt ontwikkeld rijst de vraag hoe een typisch ontwikkeltraject van mobiele applicaties wordt vormgegeven, op welke wijze dit traject verschilt van of overeenkomt met

'traditionele' softwareontwikkeling en welke keuzes er gemaakt dienen te worden tijdens de ontwikkeling van mobiele applicaties.

Mobiele apparaten worden tegenwoordig veelvuldig gebruikt voor taken die traditioneel op niet-mobiele apparaten werden uitgevoerd. Verschillende bronnen melden dat de *personal computer* zoals men die kende zal verdwijnen. De ontwikkeling van webbased software is een veelal bekend proces, maar hoe zit het proces voor de ontwikkeling van mobiele (web)applicaties in elkaar en in welke opzichten verschilt dit met het proces van de ontwikkeling van traditionele applicaties?

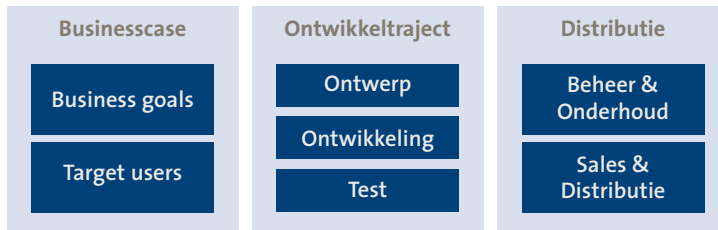
In dit artikel wordt een standaardontwikkeltraject van 'traditionele' software vergeleken met het ontwikkeltraject van mobiele applicaties, waarbij wordt verkend waar



J.P. Veldkamp MSc is junior adviseur bij KPMG Management Consulting, IT Advisory. veldkamp.jeroen@kpmg.nl



G.A. Hemmers MA is manager bij KPMG Management Consulting, IT Advisory. hemmers.gerald@kpmg.nl



Figuur 1. Stappen in een softwareontwikkeltraject.

verschillen en overeenkomsten zichtbaar zijn. De aspecten die hierbij worden besproken zijn divers en beslaan het hele traject, maar niet uitputtend; hiervoor biedt één artikel niet voldoende ruimte. Dit artikel geeft echter een goed beeld van de verschillende stappen in het proces en benoemt welke keuzes gemaakt moeten worden in de businesscase, op het gebied van het technische platform, distributiestrategie en aanpak van beheer. Dit biedt handvatten voor het aangaan van een uitdagend traject van mobiele-softwareontwikkeling.

Ontwikkeling van (mobiele) applicaties

Voor het ontwikkelen van software worden verschillende methoden gebruikt. Een bekend traditioneel model is het *Waterfall Model*, maar tegenwoordig genieten *agile* methoden grote populariteit. Een *agile* ontwikkelmethode is een methodiek waarbij er in korte iteraties bruikbare delen van de uiteindelijke applicatie worden opgeleverd (een iteratie duurt vaak twee tot drie weken). De vraag is echter of de gekozen ontwikkelmethode van invloed is op een vergelijking tussen mobiele- en traditionele-softwareontwikkeling; deze invloed is vermoedelijk klein. De stappen die men onderscheidt in de ontwikkelmethoden zijn namelijk veelal vergelijkbaar, echter de volgorde van processtappen kan verschillen. Om een goed beeld te krijgen van de verschillen in ontwikkeling voor mobiele applicaties en (web)applicaties wordt er in dit artikel nadruk gelegd op de processtappen en niet op het proces zelf.

Figuur 1 toont de standaardprocesstappen die voorkomen in het complete softwareontwikkeltraject, waarbij de driedeling gemaakt kan worden tussen de businesscase, het daadwerkelijke ontwikkeltraject en de distributie van de software. In de volgende paragrafen wordt per onderdeel een verdere analyse gedaan.

Businesscase

Voordat een applicatie wordt ontwikkeld dient er een uitgewerkte businesscase te bestaan, hoewel dit in de praktijk vaak niet het geval is. Voor het ontwikkelen van een mobiele applicatie geldt hetzelfde principe: begin niet met een ontwikkeltraject voordat de waarde voor de business zichtbaar is. Door de eigenschappen van mobiele applicaties, zoals bijvoorbeeld de doelgroep, het toepassingsgebied

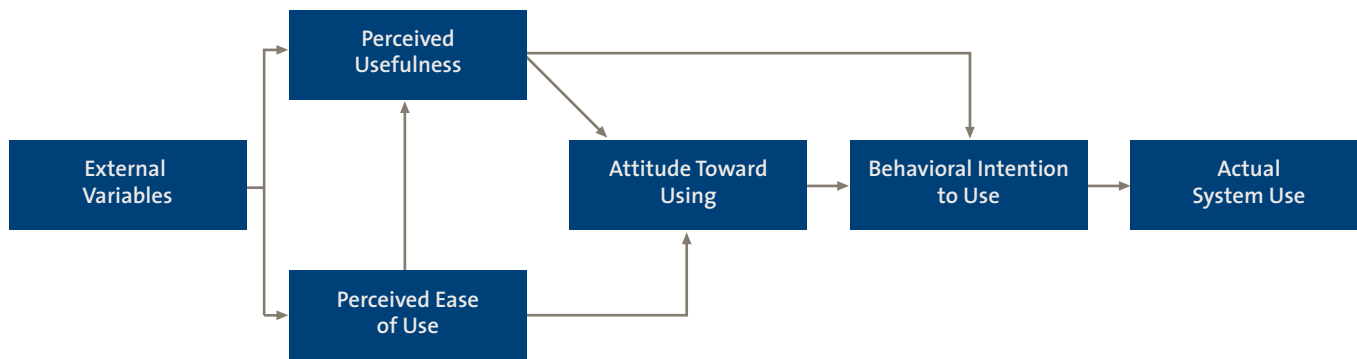
Aanvulling van een businesscase: Appie

De Albert Heijn bonuskaart bestaat al geruime tijd en is gebaseerd op een degelijke businesscase. Door middel van Albert Heijns mobiele applicatie 'Appie' is het eenvoudig om je eigen aankopen terug te zien en deze eenvoudig op te slaan in een boodschappenlijst. Daarnaast biedt Appie onder meer de mogelijkheid om, wanneer een boodschappenlijst is aangemaakt, de boodschappen efficiënter te ordenen zodat men sneller de boodschappen kan doen. De voordelen voor Albert Heijn zijn legio: het gebruik van de bonuskaart wordt gestimuleerd, de klant wordt extra gebonden en Albert Heijn heeft ook via het mobiele kanaal contact met de klant. Men kan stellen dat de bonuskaart niet meer alleen een bonuskaart wordt maar een persoonlijke winkelkaart, doordat Appie een aantal voordelen biedt aan de consument.



Verandering van de businesscase: Facebook

De mobiele Facebook-applicatie laat zien dat het implementeren van een mobiele applicatie het businessmodel van een bedrijf kan beïnvloeden. Het inkomstenmodel van Facebook is voor een groot deel gebaseerd op inkomsten uit advertenties, welke worden getoond wanneer gebruikers van Facebook ingelogd zijn. In april 2012 maakte meer dan vijftig procent van de Facebook-gebruikers gebruik van mobiele Facebook-applicaties. In deze applicaties worden echter geen advertenties getoond vanwege de beperkte schermgrootte van mobiele apparaten. Facebook is op dit moment aan het onderzoeken hoe men aan het gebruik van de mobiele producten geld kan verdienen. Verwacht wordt dat het aantal mobiele Facebook-gebruikers voorlopig nog fors zal toenemen.



Figuur 2. Het Technology Acceptance Model ([Davi89]).

en de mobiliteit, is het onwaarschijnlijk dat een bestaande businesscase in zijn geheel duplicerbaar is.

Business goals, user goals en users

Centraal bij de ontwikkeling van software staat het doel van de applicatie: wat is het doel van de software in de context van de business, wat zijn de doelen van de gebruikers en wie zijn deze gebruikers. Hoewel een helder doel in de praktijk niet altijd gesteld wordt, is dit van groot belang voor de uiteindelijke acceptatie van nieuwe software binnen de doelgroep. Het ervaren gebruikersgemak samen met het ervaren nut vormt namelijk de attitude ten opzichte van het gebruik van nieuwe technologie. Dit belang werd reeds in 1989 al omvat in het Technology Acceptance Model ([Davi89]) en vormt een basis voor andere gedragsmodellen met betrekking tot de acceptatie van technologie. Wanneer er geen helder doel of voordeel is, dan zal het nut van de software minder duidelijk zijn en zal de software minder worden geaccepteerd. Dit geldt

zowel voor mobiele als traditionele software, maar doordat de context van mobiele applicaties zeer specifiek is, zal het doel extra helder moeten zijn.

Tijdens het vormen van de businesscase moet er een algemene visie op het gehele softwareontwikkeltraject worden gevormd, omdat stappen in het proces van invloed zijn op de haalbaarheid van het traject. Hierbij moet onder meer in grote lijnen duidelijk worden welke keuzes op het gebied van ontwikkeling, distributie en doelgroep worden gemaakt. Door deze keuzes in de businesscase te omvatten kan antwoord gegeven worden op belangrijke vragen, zoals: bereik ik mijn doelgroep door de ontwikkeling voor dit platform? Welke platforms moeten ondersteund worden? Wordt de applicatie aangeboden in een app store?

Het vormen van deze visie is zowel voor de ontwikkeling van traditionele software als voor de ontwikkeling van mobiele applicaties van essentieel belang. Er zijn echter

Verschillen tussen mobiele apparaten en traditionele computers

Hoewel de nadruk in dit artikel ligt op het ontwikkeltraject voor mobiele software, is het van belang om kort stil te staan bij enkele essentiële verschillen in hardware tussen mobiele en desktopcomputers. Deze verschillen zijn namelijk in bepaalde gevallen van invloed op de ontwikkeling en toepasbaarheid van de software. Een aantal verschillen:

- Beeldschermresolutie

De meeste computers (meer dan 85 procent) hebben een beeldschermresolutie van meer dan 1024×768 pixels (W3C, 2012). De beeldschermresoluties van smartphones en tablets zijn over het algemeen lager. Voor tablets is dit echter in hoog tempo aan het veranderen, zo bevatten de laatste generaties Android-tablets en iPads resoluties van 1280 en hoger. Het grote verschil vindt men in de verscheidenheid aan beeldschermresoluties van mobiele apparaten. Een belangrijke afweging voor de ontwikkeling van mobiele applicaties is voor welke apparaten (tablet of smartphone) en welke resoluties er ontwikkeld gaat worden.

- Rekenkracht

De rekenkracht en het beschikbare werkgeheugen van mobiele apparaten is kleiner dan van een gemiddelde desktopcomputer. Dit is voor veel toepassingen geen probleem, mits er rekening mee gehouden wordt in de architectuur van de toepassing. Bepaal of de gekozen softwareoplossing voor mobiele applicaties voldoende rekenkracht biedt.

- Gebruikersinteractie

De manier waarop een gebruiker interacteert met de applicatie vindt bij een pc vaak plaats door middel van een toetsenbord of een muis. Bij mobiele apparaten is bediening door aanraking van het scherm (touch) de meest gebruikte interactiemethode. Daarnaast ziet men een toename in voice controlled interactie en het gebruik van apparaatspecifieke technieken, zoals locatie of beeld van een camera.

Gebruik van mobiele software moet een doel hebben in de specifieke context

wel verschillen tussen mobiele en traditionele applicaties als het gaat om de keuzes die gemaakt worden om tot een visie te komen.

Ontwikkeltraject

Het ontwikkeltraject van software kan grofweg worden ingedeeld in de stappen ontwerp, ontwikkeling en testen. De volgorde en het aantal herhalingen van deze stappen zal per ontwikkelmethode verschillen en in de meer *agile* ontwikkelmethoden parallel doorlopen kunnen worden.

Ontwerp

Ontwerp is een breed begrip in softwareontwikkeling; het kan slaan op een groot aantal verschillende *specialismen* van ontwerp, zoals grafisch, interactie-, functioneel, architectuur- of technisch ontwerp. Al deze 'soorten' ontwerp zijn zichtbaar in zowel mobiele als traditionele software, maar zijn veelal verschillend in invulling.

Grafisch ontwerp

De vormgeving van traditionele applicaties is veelal divers. De vormgeving van mobiele applicaties is meestal meer uniform; vanuit de verschillende platformen, zoals Android of Apple iOS, zijn er standaard interactie-elementen en designrichtlijnen. In verschillende mobiele applicaties is hierdoor een vergelijkbare *look and feel* terug te zien. Dit heeft als voordeel dat gebruikers door herkenning snel de interface van een nieuwe applicatie leren kennen. De gebruiker van een mobiele applicatie verwacht ten opzichte van desktopapplicaties altijd dat een mobiele applicatie snel en eenvoudig te bedienen is. Om hieraan te voldoen is het van belang dat het ontwerp ook eenvoudig en overzichtelijk is. Het principe '*Less is more*' is daardoor van een groter belang in mobiele applicaties dan bij desktopapplicaties.

Er is tevens een aantal technische aspecten die van invloed zijn op het grafisch ontwerp voor mobiele apparaten. Deze apparaten hebben bijvoorbeeld de mogelijkheid om te roteren. Dit zorgt ervoor dat er tijdens het grafisch ontwerp goed nagedacht dient te worden over de weergave in *landscape* en *portrait* modus en er eventueel verschillende ontwerpen gemaakt moeten worden. Daarnaast is er bijvoorbeeld een grote verscheidenheid in de pixeldichtheid (dpi) van een scherm van een mobiel apparaat, terwijl dit bij een desktopmonitor over het algemeen 72 dpi is. Het resultaat: een icoon ontworpen voor een oudere iPhone lijkt verkleind wanneer deze wordt weergegeven op een iPhone 4.

Gezien het belang van het grafische ontwerp in mobiele applicaties is het aan te raden om, in vergelijking met traditionele software, meer tijd in te ruimen voor het ontwerpen van de applicatie.

Functioneel en interactie

Door de eigenschappen van mobiele apparaten worden de mogelijke functionaliteiten en de interactiemogelijkheden beïnvloed. Het gebruik van mobiele software moet een doel hebben in de specifieke context, waardoor de functionele eisen veelal beperkter zijn dan in traditionele software. De bijkomende interactie binnen de functionaliteit zal aan moeten sluiten bij de interactie van het platform waarop de mobiele software wordt aangeboden, denk bijvoorbeeld aan het 'swipen' voor navigatiedoelinden.

Het mobiele apparaat biedt ook andere interactievormen dan een normale desktop, zoals *touch*- of spraakbesturing. Daarnaast kan gebruik worden gemaakt van verschillende inpu-telementen, zoals de beweging van het apparaat, camerabeelden en gps-coördinaten. Met de komst van de nieuwe generatie smartphones is er voor de gebruiker een belangrijke beleving bij gekomen. Het apparaat integreert content van verschillende applicaties met elkaar. Voor de gebruiker is een Facebook- of Skype-contact net zo belangrijk als een adresboekcontact en de telefoon zal deze dan ook in één lijst laten zien. Op het moment dat een mobiele applicatie hiervan afwijkt zal dit door de gebruiker als storend worden ervaren.

De verschillende interactiemogelijkheden bieden de ontwerper een grotere keuze in het ontwerp. Om de eenvoud van een applicatie te garanderen dient een ontwerper een keus te maken welke interactiemethoden de beste oplossing zijn om het doel te bereiken. Een goede interactie en functioneel ontwerp is van groot belang voor het succes van een applicatie en het loont de moeite hier extra tijd in te investeren. De gebruiker heeft, ten opzichte van traditionele software, een hogere verwachting van de gebruiksvriendelijkheid en de interactie van mobiele applicaties.

Architectuur en technisch ontwerp

Zowel bij mobiele als traditionele applicaties kan de inrichting van de architectuur achter de software zeer verschillend zijn. Afgezien van de presentatielaag in de vorm van de applicatie, zijn de mogelijke oplossingen divers. Er zijn echter binnen mobiele applicaties in het technisch ontwerp wel beperkingen, want de applicatie moet op het platform van het mobiele apparaat draaien. Net als bij de ontwikkeling van traditionele software kan ervoor worden gekozen om een webbased applicatie te bouwen, maar hierdoor kan de functionaliteit van een applicatie echter beperkt worden. Op dit moment zijn er nog maar een paar belangrijke spelers op de mobiele markt. Apple met iOS, Google met Android, BlackBerry OS (verliest veel gebrui-

Stand-alone apps	Voordelen	Nadelen	Voorbeelden
Dit zijn applicaties die geïnstalleerd moeten worden op het mobiele apparaat of de desktop-omgeving. De applicatie is volledig functioneel zonder internetverbinding en de data die gebruikt wordt in de applicatie wordt opgeslagen op het apparaat zelf.	<ul style="list-style-type: none"> • Onafhankelijk van beschikbaarheid van een internetverbinding. • Maximaal gebruik van platform- en hardware-eigenschappen. • Doordat de gebruiker een app downloadt voelt de gebruiker zich de eigenaar (feeling of ownership). • Het niet gebruiken van internetverbinding is mogelijk veiliger. • Eenvoudiger toegang tot opslag op het apparaat. 	<ul style="list-style-type: none"> • Niet mogelijk gebruik te maken van onlinegegevens. • Platformspecifieke ontwikkeling. • Vaak niet geschikt wanneer er sprake is van dynamische gegevens. • De applicatie moet meestal via tussenkomst van de platformleverancier verspreid worden (via een app store). • Uitgebreider testen. 	<ul style="list-style-type: none"> • Adresboek • Veel spelletjes • Wekkerapplicatie
Web apps	Voordelen	Nadelen	
Applicaties die binnen een browseromgeving functioneren. Deze applicaties hoeven niet geïnstalleerd te worden op het apparaat zelf. Voor deze applicaties is er echter altijd een internetverbinding nodig. Als er gebruik wordt gemaakt van de laatste webtechnieken is het mogelijk om een applicatie (in beperkte mate) te gebruiken wanneer er geen internetverbinding mogelijk is.	<ul style="list-style-type: none"> • Werkt direct zonder installatie. • Onafhankelijk van platformen. • Eenvoudig dynamische content gebruiken. • Mogelijke eenvoudiger integratie. • Geen tussenkomst van platformleverancier. • Geen distributiekosten. • Relatief eenvoudig testtraject. • Uitvoeren van updates zonder dat er een nieuwe applicatie wordt geïnstalleerd. 	<ul style="list-style-type: none"> • Geen internet? Geen werkende applicatie! • Meer aandacht aan beveiliging. • Niet geschikt om client-side zware berekeningen uit te voeren. • Geen feeling of ownership doordat de app niet wordt gedownload. 	<ul style="list-style-type: none"> • YouTube • Gmail.com • Amazon
Hybride apps	Voordelen	Nadelen	
Applicaties die geïnstalleerd dienen te worden op een apparaat, maar vaak content of interactie-elementen via het internet ophalen. Hierdoor is er voor het gebruik van de applicatie vaak toch een internetverbinding nodig.	<ul style="list-style-type: none"> • Blijft (deels) functioneel wanneer internetverbinding niet beschikbaar is. • Feeling of ownership. • Dynamisch laden van (deel van) gegevens. • De snelheid van een standalone app. • Content eenvoudig te updaten zonder tussenkomst van de gebruiker. 	<ul style="list-style-type: none"> • Platformspecifieke ontwikkeling. • Installatie vereist. • Tussenkomst van platformleverancier. • Uitgebreider testen. • Gegevens op twee plaatsen (onlineserver en in app). 	<ul style="list-style-type: none"> • Facebook app • Gmail app • BNR app • Buienradar

Tabel 1. Een aantal voor- en nadelen van typen applicaties.

kers aan IOS, Google en Windows) en Windows mobile (wint aan marktsegment maar blijft op dit moment nog ver achter bij de andere spelers).

Het is aan te raden om alvorens de applicatieontwikkeling start goed na te denken voor welke platformen minimaal ontwikkeld wordt. Houd hierbij rekening met de specifieke doelgroep van de mobiele applicatie en onderzoek welke platformen er door de doelgroep gebruikt worden. Deze keuze is van belang, omdat die invloed heeft op het hele ontwikkeltraject.

Ontwikkeling

Voor de ontwikkeling van de software dient er, op basis van de *requirements* die indirect volgen uit de businesscase, een keuze gemaakt te worden welk type software ontwikkeld wordt. De keuze voor de ontwikkeling van software uit één van de volgende categorieën is onder meer afhankelijk van de eisen en het doel van de applicatie. Bij de ontwikkeling van software kan hierin onderscheid gemaakt worden in drie hoofdcategorieën, namelijk *stand-alone* (ook wel *dedicated*), *web* en *hybride* applicaties. Elke categorie

heeft haar eigen voordelen, nadelen en specificaties. In tabel 1 wordt een aantal voor- en nadelen van de verschillende categorieën uiteengezet.

De keuze voor het type applicatie heeft grote invloed op het ontwikkeltraject en met name op de mogelijke ontwikkelmethoden. Bij een (mobiele) webapplicatie hoeft de ontwikkelaar geen rekening te houden met het platform waarop de applicatie komt te draaien, zolang de applicatie goed werkt via een browser. Maar let op: de verschillende browsers van de verschillende platformen kunnen de content wel verschillend weergeven. Wordt er echter een applicatie ontwikkeld waarbij een installatie op het apparaat vereist is, dan moet de software ontwikkeld worden in een programmeertaal die door het platform wordt ondersteund. Voor het ontwikkelproces begint, moet duidelijk zijn welke (versies van) platformen ondersteund moeten worden.

De hardware in de mobiele apparaten is zeer divers

Voor de ontwikkeling van mobiele applicaties is het mogelijk interessant om gebruik te maken van *frameworks* waarin *stand-alone* applicaties ontwikkeld kunnen worden, zonder te ontwikkelen in de programmeertaal van het platform waarop de applicatie gaat draaien. Hiermee is het mogelijk om in één programmeertaal een applicatie te ontwikkelen, om deze vervolgens door het framework om te laten zetten (compileren) naar verschillende platformen, bijvoorbeeld iOS, BlackBerry, Windows en Android. De kwaliteit, mogelijkheden en prestaties van deze frameworks zijn echter zeer divers, dus onderzoek goed of een framework aansluit bij de eisen van de applicatie.

Test

Het testen van software is een belangrijk en essentieel onderdeel van het ontwikkeltraject. Dit geldt voor alle vormen van softwareontwikkeling. Het soort software (zie tabel 1) dat wordt ontwikkeld is van grote invloed op het testtraject, voornamelijk wanneer er gekozen wordt voor een applicatie waarbij een installatie op een apparaat nodig is.

Het aantal versies en het verloop van mobiele besturingsystemen is groot, de beeldschermresoluties variëren sterk en de hardware in de mobiele apparaten is zeer divers, waardoor er een groot aantal combinaties van hardware en software ontstaat. Dit maakt het complex om goede tests te ontwikkelen. Wanneer er gekozen wordt voor het ontwikkelen van een applicatie die binnen een browser functioneert, dan hoeft men alleen rekening te houden met de eigenschappen van de verschillende webbrowsers, waardoor het testen minder complex wordt.

Het is de moeite waard om te onderzoeken of het voordelig is om het uitvoerig testen van de applicatie uit te besteden. Wanneer het aantal apparaten dat u wilt ondersteunen divers is, dan is het in de praktijk niet eenvoudig om daadwerkelijk tests uit te voeren op de verschillende apparaten. Er zijn bedrijven die hierin gespecialiseerd zijn en het testen voor u uit kunnen voeren. Afhankelijk van de

Mogelijkheden onduidelijk? Doe een Proof of Concept (PoC)!

Bestaat er tijdens het traject onduidelijkheid over de mogelijkheid om aan bepaalde (functionele) eisen te voldoen, voer dan gedurende het traject verschillende Proof of Concepts uit. Deze tests geven vaak een erg goed beeld van de mogelijkheden en beperkingen van de uiteindelijke oplossing. Wacht vooral niet met het uitvoeren van een test tot het te laat is. Het uitvoeren van PoC's is voor zowel traditionele als mobiele applicaties aan te raden.

complexiteit van de applicatie en de diversiteit aan mobiele apparaten waarop de applicatie gebruikt gaat worden, kan dit kostenefficiënt zijn.

Aangezien het testen van software voor mobiele apparaten nog redelijk in de kinderschoenen staat is er een kleinere keus aan softwarematige testoplossingen. Van het geïntegreerd testen van de applicatie tijdens de ontwikkeling is nog nauwelijks sprake. Op dit moment zijn er frameworks die geïntegreerd kunnen worden in een mobiele applicatie zodat er inzicht ontstaat over het gebruik van de applicatie. Daarnaast zenden veel gebruikers zogenaamde debuginformatie door wanneer een applicatie vastloopt. Deze twee methoden geven veel inzicht in eventuele fouten in de mobiele software maar zijn afhankelijk van de interactie van de eindgebruiker en verlengen de testperiode aanzienlijk.

Verschillen in het testtraject tussen mobiele applicaties en traditionele applicaties worden voornamelijk veroorzaakt door de hardwarematige diversiteit van mobiele apparaten. Een groot deel van de testcomplexiteit vervalt wanneer er een mobiele webapplicatie (een webapp) wordt ontwikkeld die geen gebruik maakt van specifieke hardwarefuncties, maar alleen draait in de webbrowser van het mobiele apparaat. Hierdoor hoeft er alleen rekening gehouden te worden met de browsers, wat voor traditionele applicaties ook al het geval is.

Distributie

Na de ontwikkeling moet de software worden verspreid, aangeboden aan de doelgroep en moet er een mechanisme ontwikkeld worden om in beheer en onderhoud te voorzien. Bij de ontwikkeling van webapplicaties hoeft de applicatie niet los gedistribueerd te worden of bij de gebruiker geüpdatet te worden, omdat deze vanuit een browser kan worden geopend. Voor het distribueren van (dedicated en hybride) software zijn er verschillen tussen de distributie van desktopapplicaties en mobiele applicaties zichtbaar.

Het distribueren van desktopsoftware is zeer eenvoudig geworden, doordat het mogelijk is om software ter download aan te bieden aan de gebruiker via onlinewinkels of een website. Hierdoor is het gebruik van cd's, dvd's of andere media voor distributie in sterke mate afgenomen, maar er kan ook nog gekozen worden voor de distributie via fysieke winkels.

Bij de distributie van mobiele software moet er (bijna) altijd gebruik worden gemaakt van de online ‘winkel’ van de leverancier van het platform waarop de applicatie moet draaien. Hierbij moet rekening gehouden worden met de voorwaarden van deze leverancier, zoals controle op applicatiekwaliteit of provisies over verkoop van de applicatie. Er zijn verschillen tussen mobiele licenties van verschillende leveranciers, deze zijn in kaart gebracht in tabel 2.

Beheer

Het beheerproces van een applicatie is onder te verdelen in het technisch beheer, het functioneel beheer en het applicatiebeheer. De beheeromgeving van een mobiele webapplicatie zal niet verschillen van die van een normale webapplicatie. Op het moment dat de beheerorganisatie ingericht wordt voor hybride of stand-alone applicaties kunnen deze gebruikmaken van dezelfde processen, omdat het verschil zit in de technische oplossing.

Aangezien de kans groter is bij een mobiel apparaat dat dit vergeten of gestolen wordt, is het nog belangrijker om ervoor te zorgen dat de data van een apparaat en de applicaties beveiligd worden. Dit soort beheermaatregelen worden samengevat onder de noemer Mobile Device Management (MDM). MDM-oplossingen bieden over het algemeen de volgende beheermaatregelen:

- op afstand het mobiele apparaat blokkeren;
- op afstand kunnen leegmaken van het apparaat;

- uitvoeren van back-ups;
- traceren van het apparaat;
- afdwingen van extra beveiligingscertificaten;
- uitvoeren van software-updates.

Op dit moment zijn er tal van leveranciers die oplossingen aanbieden om MDM in te richten. Alvorens men begint met de ontwikkeling van een mobiele applicatie is het van belang goed stil te staan bij de vraag op welke wijze MDM ingericht dient te worden.

Een ander belangrijk aspect van de software lifecycle is de doorontwikkeling van de software. Pas na de distributie van de applicatie komen er vaak nieuwe wensen vanuit de gebruikers. Hierin ziet men geen verschil tussen de ontwikkeling van reguliere en mobiele software. Het verschil doet zich voor in de wijze waarop de gebruikers feedback geven. Doordat men gebruikmaakt van een mobiele telefoon is men eerder geneigd om zowel positieve als negatieve bevindingen van een mobiele applicatie direct te delen via verschillende social mediakanalen. Het is noodzakelijk om deze kanalen te gebruiken om een completer inzicht te krijgen in de wensen van de gebruikers.

Als er gekeken wordt naar de release cycle van software dan is er een verschil zichtbaar in de doorlooptijd voor de ingebruikname van een nieuwe versie van de mobiele applicatie. De meeste gebruikers updaten niet gelijk wanneer er een nieuwe versie van de app geïnstalleerd wordt.

Omschrijving	Apple	Android	BlackBerry	Windows
Distributiemogelijkheden via een centrale locatie	App store	Play store	Market	Marketplace
Acceptatie controle door de leverancier	Ja	Nee	Nee	Ja
Vrije distributie van app zonder tussenkomst van de leverancier	Nee *	Ja	Ja	Ja
Kosteloze licentie voor verspreiding van de applicatie	Nee **	Ja	Nee ***	Ja
Verplichte afdracht voor de verkoop van producten binnen de app	30%	30%	Nee	30%****
Verkoop van de app buiten de app store mogelijk	Nee *	Ja	Ja	Ja
Afdracht voor verkoop van de app via de app store	30%	30%	Nee	30%

Tabel 2. Verschillen tussen app stores van leveranciers.

* Apple biedt wel de mogelijkheid om via een coöperatielicentie een applicatie te verspreiden binnen een bedrijf. Hiervoor dient de licentie wel toegekend te worden aan een bedrijf waarbij een KvK-nummer tot de verplichtingen behoort.

** Apple werkt met jaarlicenties; afhankelijk van de gekozen licentie betaalt men \$99 of \$299 per jaar. Op het moment dat men zowel een applicatie wil verspreiden via de app store als via het eigen intranet dienen er twee licenties aangeschaft te worden.

*** BlackBerry vraagt per tien applicaties een ontwikkellicentie van \$200.

**** Op het moment dat de verkoop boven de \$25.000 stijgt wordt er 20% berekend.

Om het gedrag van de gebruikers positief te beïnvloeden kan er bijvoorbeeld nagedacht worden over het vanuit de app zelf aankondigen van nieuwe updates. Er dient van tevoren goed nagedacht te worden over hoe hiermee omgegaan wordt aangezien dit gevolgen kan hebben voor het ontwikkelproces.

Conclusie

In dit artikel is een standaard-softwareontwikkeltraject uiteengezet, waarbij is gekeken naar verschillen en overeenkomsten tussen het ontwikkelen van traditionele applicaties en mobiele applicaties. Hieruit blijkt dat de stappen die in het proces voorkomen vergelijkbaar zijn, maar dat er in de invulling van de stappen een aantal duidelijke verschillen zichtbaar is. Kernpunten hierin zijn onder meer:

- Door de eigenschappen van mobiele applicaties, zoals bijvoorbeeld de doelgroep, het toepassingsgebied en de mobiliteit, is het onwaarschijnlijk dat een bestaande businesscase in zijn geheel dupliceerbaar is. Onderzoek de mogelijkheden.
- De gebruiker van een mobiele applicatie verwacht ten opzichte van desktopapplicaties altijd dat een mobiele applicatie snel en eenvoudig te bedienen is. Om hieraan te voldoen is het van belang dat het ontwerp ook eenvoudig en overzichtelijk is: less is more!
- Het gebruik van mobiele software moet een doel hebben in de specifieke context, waardoor de functionele eisen veelal beperkter zijn dan in traditionele software. Door de eigenschappen van mobiele apparaten worden de mogelijke functionaliteiten en de interactiemogelijkheden beïnvloed.
- De keuze voor het type applicatie heeft grote invloed op het ontwikkeltraject en met name op de mogelijke ontwikkelmethoden.
- Voor de ontwikkeling van mobiele applicaties is het mogelijk interessant om gebruik te maken van een framework waarin dedicated stand-alone applicaties ontwikkeld kunnen worden, zonder te ontwikkelen in de programmeertaal van het platform waarop de applicatie gaat draaien.

- Het is de moeite waard om te onderzoeken of het voordelig is om het uitvoerig testen van een mobiele applicatie uit te besteden.
- Bij de distributie van mobiele software moet er vaak gebruik worden gemaakt van de online ‘winkel’ van de leverancier van het platform waarop de applicatie moet draaien. Houd rekening met de beperkingen en kosten.
- Besteed aandacht aan het inrichten van een Mobile Device Management-oplossing en de release cycle.

Kortom: er zijn verschillen en aandachtspunten bij het ontwikkelen van mobiele applicaties, maar zij vormen leuke uitdagingen ten opzichte van het ontwikkeltraject van traditionele software, en bieden juist volop kansen.

Literatuur

- [**Davis89**] F. Davis, *Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology*. MIS Quarterly 13 (3), 1989, p. 319-340.
- [**Horn11**] L. Horn (2011, Oktober 24). 1. Retrieved Mei 20, 2012, from Report: Android Market Reaches 500,000 Apps: <http://www.pcmag.com/article2/0,2817,2395188,00.asp>.
- [**IGFK11**] Intomart GFK, *Trends in Digitale Media*. Intomart GFK, 2011.
- PCMag (n.d.)**. *Report: Android Market Reaches 500,000 Apps*. Retrieved 03 10, 2012, from PC Magazine: <http://www.pcmag.com/article2/0,2817,2395188,00.asp>.
- W3C (n.d.)**. *Browser Display Statistics*. Retrieved 06 02, 2012, from W3C Schools: http://www.w3schools.com/browsers/browsers_display.asp.

Over de auteurs

- J.P. Veldkamp MSc** is junior adviseur bij KPMG Management Consulting, IT Advisory. In zijn dagelijks werk ontwerpt en ontwikkelt hij (mobiele) applicaties binnen verschillende innovatieve *agile* projecten. Hij heeft een achtergrond in Information Science met een focus op business information systems en e-commerce.
- G.A. Hemmers MA** is manager bij KPMG Management Consulting, IT Advisory, en heeft meer dan twaalf jaar ervaring met het ontwerpen van innovatieve webapplicaties. In zijn dagelijks werk is hij verantwoordelijk voor onder andere mobiele app-ontwikkelingen en prototyping binnen KPMG. Hij heeft een achtergrond als userinterface designer en creëert eenvoudige oplossingen voor complexe materie.