

# Een vergelijkende studie van cross-platform tools voor het ontwikkelen van mobiele applicaties

Michiel Staessen

Departement computerwetenschappen

KU Leuven

michiel.staessen@student.kuleuven.be

## Samenvatting

*Het ontwikkelen van mobiele applicaties voor meerdere platformen is een tijdrovende en dure bezigheid. Omwille van deze reden zoeken meer en meer bedrijven hun toevlucht in cross-platform tools (CPT's) voor het ontwikkelen van mobiele applicaties (of apps). In opdracht van Capgemini, een internationaal consultingbedrijf, presenteert dit artikel een methode voor het evalueren en selecteren van cross-platform tools in een bedrijfsomgeving. Aansluitend worden de resultaten gepresenteerd van de toepassing van deze methode op de native SDK's en twee cross-platform tools, Apache Cordova of PhoneGap en Motorola Rhodes. De tools worden getest op hun inzetbaarheid voor het maken van smartphone en tablet applicaties voor Android en iOS.*

## 1. Inleiding

Smartphones zijn razend populair en de verkoop ervan is op zijn minst indrukwekkend: sinds 2009 hebben de fabrikanten een samengestelde jaarlijkse groei van 57,61% weten te realiseren, uitgaande van wereldwijd marktonderzoek door Gartner [1]–[16]. Deze groei wordt voornamelijk aangedreven door goedkope Android toestellen die de plaats innemen van de traditionele GSM.

Een soortgelijk scenario speelt zich af op de markt van de tablets. Volgens een schatting van IDC zullen tablet fabrikanten tussen 2012 en 2016 wereldwijd een samengestelde jaarlijkse groei van 23,3% realiseren [17].

De huidige smartphonemarkt wordt gedomineerd door het Apple/Google duopolie. Van de 700 miljoen verkochte smartphones is 85% een Android toestel of iPhone. Door netwerkeffecten is het erg moeilijk voor andere platformen om marktaandeel te verkrijgen. Desalniettemin blijven ontwikkelaars naar alternatieven zoeken omdat ze als de dood zijn voor vendor lock-in [18].

Cross-platform tools (CPT's) laten ontwikkelaars toe om applicaties te ontwikkelen voor meerdere platformen tegelijk, vertrekkende van een (quasi) identieke codebase. CPT's verlagen daarmee zowel toetredings- als uittredingsdrempels ("lock-in"). Door een CPT te gebruiken, krijgt de ontwikkelaar eenvoudig toegang tot nieuwe platformen en kan de ontwikkelaar gemakkelijker naar een ander platform migreren [19].

CPT's proberen een oplossing te bieden voor drie problemen [19]:

- Het eerste doel is uiteraard het overwinnen van platform fragmentatie want de apparaten zijn op zeer diverse vlakken erg verschillend: besturingssysteem, versie van het besturingssysteem, schermgrootte, schermresolutie, processorkracht, ...
- Daarnaast vereenvoudigen CPT's ook de toegang tot nieuwe platformen en schermen. Hoewel op dit ogenblik CPT's voornamelijk gebruikt worden voor het ontwikkelen van smartphone en tablet applicaties, kunnen CPT's ook gebruikt worden voor het ontwikkelen van applicaties voor desktops, TV's en andere media.
- Tot slot kan het gebruik van CPT's leiden tot een efficiënter beheer van ontwikkelingsmiddelen. Een applicatie voor meerdere platformen ontwikkelen impliceert namelijk dat (vaak verschillende) teams meerdere codebases en functionaliteit moeten synchroniseren over de verschillende platformen. Dit is een tijdrovende en dure bezigheid.

## 2. Mobiele architecturen

Er zijn tot op heden vijf architecturen gekend om applicaties (of apps) op meerdere platformen beschikbaar te maken: native apps, mobile web apps, hybrid apps, interpreted apps en cross-compiled apps [20].

## 2.1. Native apps

Native apps zijn de de facto architectuur voor het ontwikkelen van een mobiele applicatie en vormen dus de baseline voor de vergelijking van de architecturen.

Native apps worden ontwikkeld met de SDK die aangeleverd wordt door de ontwikkelaar van het platform. Voor Android worden native applicaties geschreven in Java, voor iOS worden deze applicaties geschreven in Objective-C. Daarenboven hebben native apps een uiterlijk dat consistent is met het uiterlijk van het besturingssysteem (ook native look & feel genoemd) en kunnen ze het volledige potentieel van het toestel benutten waardoor ze het meest performant zijn.

## 2.2. Mobile web apps

Mobile web apps zijn mobiele websites en worden geopend in de web browser van het mobiele toestel. Omdat elk platform over een browser beschikt en omdat de HTML-specificatie platform onafhankelijk is, zijn mobiele web apps de meest eenvoudige manier om voor meerdere platformen te ontwikkelen. Er zijn echter ook nadelen verbonden aan mobiele web apps.

Mobiele web apps worden verspreid met behulp van een URL en kunnen dus niet gevonden worden in applicatiewinkels, zoals de App Store en Google Play. Hoewel de HTML5 specificatie een zekere toegang verleent tot het apparaat, kunnen niet alle functies van het mobiele apparaat aangesproken worden. Bovendien is de ondersteuning van dergelijke API's vaak gebrekkig in mobiele browsers [21]. Ook de performantie laat vaak te wensen over.

Omdat de mobiele web app in de browser draait, kan deze ook geen gebruik maken van native UI elementen en kan deze dus ook geen native look & feel aanbieden. Of dit een nadeel is, staat nog ter discussie. Websites voldoen aan de “one size fits all” filosofie en mogen als dusdanig een eigen look & feel hebben [22].

Cross-Platform Tools voor web app ontwikkeling zijn HTML5 frameworks waaronder jQuery Mobile, Sencha Touch, KendoUI, Zepto.js, en DHTMLX Touch.

## 2.3. Hybrid apps

Hybrid apps zijn de combinatie van native apps en mobiele web apps. De mobile website wordt verpakt in een native omhulsel en dit omhulsel kan extra functionaliteit openstellen naar de applicatie indien de browser bepaalde functionaliteit mist. De communicatie tussen de applicatie en het omhulsel verloopt via een JavaScript bridge. Op deze manier kunnen hybrid apps

vooral nog het volledige potentieel van het mobiele apparaat benutten.

Voor de eindgebruiker is er weinig verschil tussen een hybrid apps en native apps omdat deze apps via de gewoonlijke kanalen te verkrijgen zijn (applicatiewinkels). Hybrid apps hebben geen native uiterlijk omdat ze doorgaans geen native UI elementen kunnen gebruiken voor de user interface. Hybrid apps zijn sterk afhankelijk van de browser en dus laat ook de performantie van hybrid apps vaak te wensen over.

Ondanks de nadelen, zijn hybrid apps toch interessant en in het bijzonder voor web ontwikkelaars die de overstap maken naar mobiele applicaties. Zij kunnen op deze manier hun kennis van webtechnologie gebruiken om mobiele applicaties te maken. Omdat de applicatie zelf in HTML geschreven is, laat dit toe om updates te verspreiden zonder tussenkomst van de applicatiewinkel en ondersteunen CPT's voor hybrid apps doorgaans meer platformen.

Cross-Platform Tools die dit patroon volgen zijn onder meer Apache Cordova, Sencha Cmd, Trigger.io en Kony

## 2.4. Interpreted apps

Interpreted apps maken gebruik van een tussenliggende taal die wordt vertaald in native instructies at runtime. Dit is min of meer vergelijkbaar met de Java VM die Java bytecode omzet in machinecode. Deze architectuur vereist uiteraard wel dat de runtime-omgeving meegeleverd wordt met de applicatie. Hierdoor neemt de applicatie meer ruimte in beslag op het apparaat.

De performantie van interpreted apps is sterk afhankelijk van de gebruikte tussentaal en vertaler maar is doorgaans beter dan web of hybrid apps. Omdat de interpreted apps vertaald worden naar platformspecifieke code, kunnen ze wel gebruik maken van native UI elementen en zullen ze de native look & feel kunnen aanbieden. Interpreted apps kunnen verspreid worden via app stores omdat ze — net als hybride apps — verpakt worden als native apps.

Cross-platform tools die deze architectuur omarmen zijn onder andere Appcelerator Titanium en Rhodes.

## 2.5. Cross-compiled apps

Cross-compiled apps maken — net zoals interpreted apps — gebruik van een tussenliggende taal. Alleen worden de instructies in deze taal niet tijdens het uitvoeren maar tijdens het compileren vertaald. Bijgevolg hoeft er geen runtime meegeleverd te worden met de applicatie.

De performantie is afhankelijk van de gebruikte cross-compiler maar doorgaans gelijkaardig aan de performantie van native apps. Ook de native UI elementen kunnen gebruikt worden waardoor de applicatie een vertrouwd uiterlijk heeft.

Xamarin and MoSync zijn cross-platform tools die deze architectuur gebruiken.

### 3. Methodologie

De methodologie die gebruikt wordt in dit onderzoek is gebaseerd op een generische software selectie methode, voorgesteld door Jadhav en Sonar [23]. Deze methode bestaat uit zes aanpasbare stappen die in de volgende alinea's besproken worden.

#### 3.1. Definieer selectiecriteria

In de eerste stap worden de selectiecriteria bepaald. Deze criteria zijn de functionele en non-functionele vereisten waaraan een CPT moet voldoen om in aanmerking te komen voor verdere evaluatie. Omdat dit onderzoek wordt gevoerd in opdracht van consultingbedrijf Capgemini, ligt de focus op applicaties voor de bedrijfsweld. De selectiecriteria werden bepaald op de startvergadering en houden in dat:

- de CPT applicaties kan maken voor zowel Android als iOS,
- de CPT applicaties kan maken voor zowel smartphones als tablets,
- de CPT bruikbaar is voor het maken van business applicaties,
- de CPT een geschikte maturiteit heeft,
- de CPT een goede ondersteuning heeft door de ontwikkelaar of de gemeenschap.

#### 3.2. Identificeer mogelijke kandidaten

In de deze stap wordt gezocht naar mogelijke, bruikbare kandidaten. Tijdens deze zoektocht worden de selectiecriteria nog niet in rekening gebracht. Deze fase is namelijk louter een ontdekkingsfase. Tijdens het onderzoek werden 160 mogelijke tools geïdentificeerd uit een in 2012 voltooide studie van VisionMobile [19] en een nog onvoltooide studie van research2guidance [24].

#### 3.3. Selecteer kandidaten voor evaluatie

In de derde stap worden de selectiecriteria uit de eerste stap gebruikt om de lijst van kandidaten, verkre-

gen in de tweede stap, te filteren. Uit de lijst van overgebleven tools, werd voorrang verleend aan 7 tools omdat deze ofwel erg populair zijn of specifiek ingezet worden voor het ontwikkelen van applicaties in de bedrijfsweld. Uit deze 7 tools werden twee tools geselecteerd, namelijk Apache Cordova en Motorola Rhodes. Cordova werd geselecteerd voor haar populariteit, Rhodes omwille van zijn bedrijfscontext.

#### 3.4. Definieer evaluatiecriteria

Tijdens deze stap worden de criteria waarop de alternatieven beoordeeld worden, vastgelegd. Hiervoor werd gekeken naar de literatuur [19], [25] en er werden interviews afgenomen bij een ontwikkelaar, een architect en een verkoper van Capgemini. Er zijn 11 evaluatiecriteria geformuleerd en deze zijn georganiseerd in een boomstructuur met drie hoog-niveau categorieën om de vergelijking eenvoudiger te maken. De literatuur suggereert namelijk dat mensen slechts in staat zijn om rekening te houden met  $7 \pm 2$  stukken informatie [26]. De evaluatiecriteria zijn:

- **Portabiliteit** meet het vermogen van een tool om bepaalde facetten van de software of het ontwikkelingsproces te hergebruiken voor de verschillende platformen.
  - **Platformondersteuning** meet het aantal ondersteunde platformen, naast Android en iOS.
  - **Hergebruik van tools** meet in hoeverre het mogelijk is om de ontwikkelomgeving te hergebruiken.
  - **Hergebruik van code** meet hoeveel code van een applicatie hergebruikt kan worden tussen platformen onderling.
- **Gebruikerservaring van de applicatie** gaat na in hoeverre een cross-platform applicatie kan tippen aan een native applicatie.
  - **Integratie met het platform** test de mogelijkheid om een applicatie te integreren met het platform.
    - \* **Toegang tot de hardware** gaat op zoek naar de mogelijkheden tot integratie met de hardware.
    - \* **Integratie met platform-specifieke diensten** gaat op zoek naar integratiemogelijkheden met services zoals push-notificaties.

- **Gebruikersinterface** onderzoekt de mogelijkheden van de grafische gebruikersinterface.
  - \* **Native look & feel** gaat na of de tool gebruik kan maken van de native gebruikersinterface elementen.
  - \* **Gebruikersinterface elementen** bestudeert de mogelijkheid om een rijke gebruikersinterface aan te bieden met de elementen die voorhanden zijn.
- **Performantie** beschrijft de snelheid waarmee de applicatie reageert op invoer.
- **Productiviteit** beschrijft de efficiëntie bij het maken van applicaties bij het gebruik van een zekere tool.
  - **Hergebruik van vaardigheden** kijkt naar de mogelijkheden om reeds verworven vaardigheden te hergebruiken.
  - **Gereedschap** focust op de kwaliteit van het aangeboden gereedschap om de applicatie te maken.
  - **Testen** onderzoekt de mogelijkheden van de tool om geautomatiseerde testen te schrijven.

Merk op dat deze criteria geen kosten bevatten. De kosten en baten zijn met opzet van elkaar gescheiden gehouden zodat op het einde van de vergelijking een aparte kosten-batenanalyse kan worden gemaakt.

### 3.5. Evalueer de alternatieven

Met de evaluatiecriteria uit de vorige stap, kunnen de alternatieven nu geëvalueerd worden. Jadhav en Sonar [23] dwingen geen specifieke techniek af voor deze evaluatie maar laten de keuze over aan de onderzoeker. Voor dit onderzoek werd gekozen voor het analytisch hiërarchisch proces (AHP), ontworpen door Thomas L. Saaty [27], [28]. Deze methode berekent, op basis van het eigenwaardenprobleem, gewichten of scores voor zowel evaluatiecriteria en alternatieven. Deze gewichten zijn afkomstig uit paarsgewijze vergelijkingen waarbij gebruik gemaakt kan worden van zowel kwalitatieve als kwantitatieve evaluatiecriteria. Bovendien geeft de methode een indicatie van de hoeveelheid inconsistentie die aanwezig is in het gegeven oordeel. Naast het feit dat deze methode meermaals succesvol is toegepast in de literatuur, zijn dit de voornaamste redenen voor het selecteren van deze methode.

In de eerste fase worden de evaluatiecriteria gewogen. In de tweede fase worden alle alternatieven aan een vergelijking onderworpen en worden voor elk criterium

gewichten toegekend aan de alternatieven. In de laatste fase worden alle gewichten gecombineerd door de som te berekenen van het product van een gewicht voor een bepaald alternatief met betrekking tot een bepaald criterium en het gewicht van dat criterium (additieve weging). De alternatieven kunnen dan gerangschikt worden op basis van de grootte van hun totaalgewicht.

### 3.6. Selecteer het beste alternatief

In de laatste stap, wordt het beste alternatief gekozen. De tool die het hoogste scoort in de evaluatie is daarbij niet noodzakelijk ook het beste alternatief. Bij het kiezen van het beste alternatief dienen de voordelen en de kosten tegenover elkaar afgewogen worden want geen enkel bedrijf zal haar middelen verspelen aan een inefficiënte tool. De kosten worden in dit onderzoek voorgesteld door de tijd die nodig is om de applicatie te ontwikkelen. Een applicatie bestaat namelijk uit een aantal lijnen code die in een bepaalde tijd moeten geschreven worden door een aantal mensen die moeten betaald worden.

## 4. Geselecteerde cross-platform tools

### 4.1. Apache Cordova

Apache Cordova<sup>1</sup>, beter gekend als PhoneGap, is een open-source CPT die toelaat hybride applicaties te ontwikkelen voor Android, iOS, BlackBerry, WebOS, Windows 8, Windows Phone, Symbian, Tizen en Bada.

De tool maakt gebruik van HTML5, JavaScript en CSS voor de ontwikkeling van applicaties. Met behulp van een JavaScript bridge kan de applicatie toegang krijgen tot het platform. Alle device API's zijn beschikbaar in de vorm van plug-ins. Ontwikkelaars kunnen ook hun eigen plug-ins maken indien ze specifieke native code op het platform willen aanroepen vanuit de HTML5 applicatie.

Cordova heeft een Apache 2.0 licentie en wordt beheerd door de Apache Software Foundation. PhoneGap is een distributie van Cordova, geleverd door Adobe [29]. Adobe biedt ook een clouddienst aan voor het compileren van PhoneGap applicaties.

### 4.2. Motorola Rhodes

Motorola Rhodes<sup>2</sup> is eveneens een open-source CPT die toelaat interpreted/hybride applicaties te maken voor Android, iOS, BlackBerry, Windows Mobile

<sup>1</sup><http://cordova.apache.org>

<sup>2</sup><http://docs.rhmobile.com/rhodes/introduction>

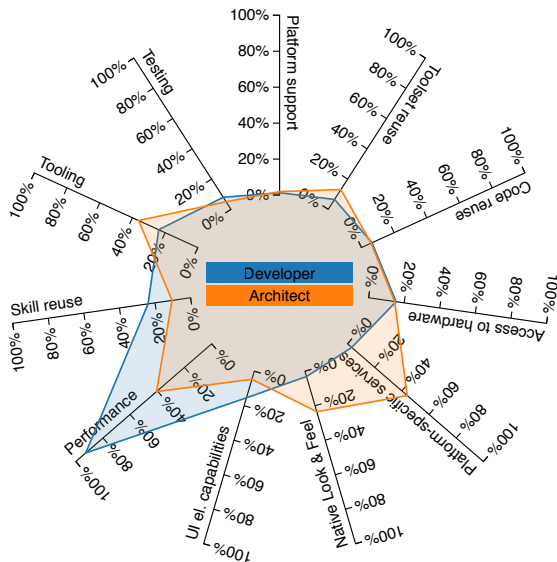
en Windows Phone 8.

De runtime omgeving van Rhodes is een Ruby-gebaseerd MVC framework voor web applicaties. De applicatie is dus een website (vergelijkbaar met Cordova) die op een lokale Ruby server op het apparaat zelf draait.

Rhodes wordt ontwikkeld door Motorola Solutions en is vrij beschikbaar onder de MIT licentie. Een online dienst voor het compileren van Rhodes projecten is eveneens beschikbaar.

## 5. Evaluatie

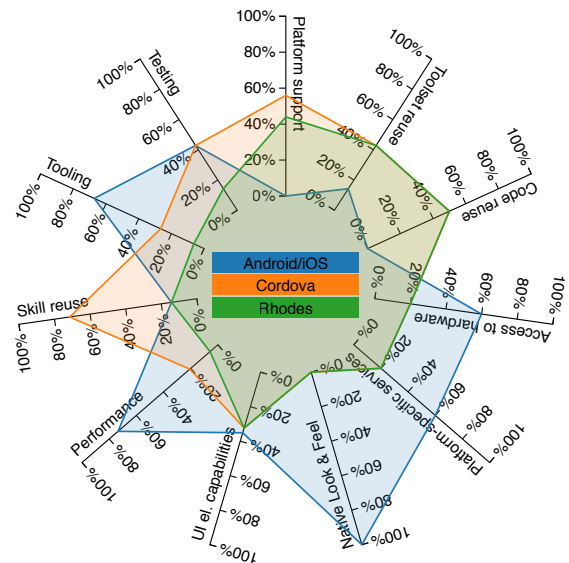
Zoals gezegd worden eerst de criteria gewogen. Dit is gebeurd door de meningen over het relatieve belang van de criteria aan een ontwikkelaar en architect te vragen in de vorm van een enquête. Deze meningen werden met behulp van de AHP methode omgezet in gewichten. De verdeling van de gewichten wordt gevisualiseerd in Figuur 1.



**Figuur 1. Overzicht van de gewichtsverdeling van de evaluatiecriteria voor de ontwikkelaar en architect.**

Voor het vergelijken van de alternatieven (de tweede fase) wordt deels gebruik gemaakt van de beschikbare documentatie en deels gebruik gemaakt van ervaring die verworven werd tijdens het ontwikkelen van een proof-of-concept applicatie met elk van de tools. Deze ervaring is nodig om een waarheidsgetrouw oordeel te kunnen vellen over de inzetbaarheid van de tools in een bedrijfsomgeving en de proof-of-concept applicatie combineert een aantal elementen die typisch voorkomen in applicaties voor deze markt. De gewich-

ten van de alternatieven worden gevisualiseerd in Figuur 2.



**Figuur 2. Overzicht van de gewichtsverdeling bij de alternatieven.**

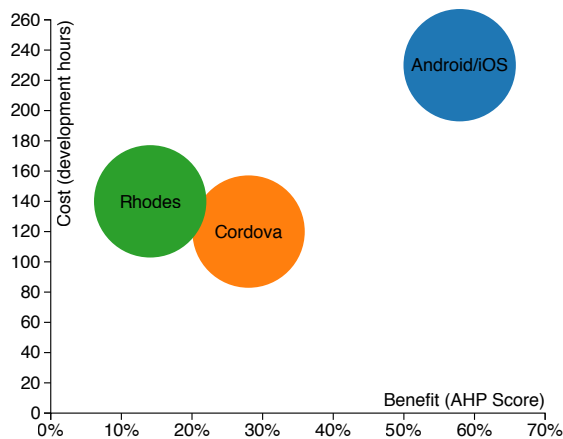
Vervolgens worden deze scores gecombineerd. De totaalgewichten worden weergegeven in Tabel 1. Omdat de meningen uiteenlopend zijn, wordt de evaluatie afzonderlijk uitgevoerd vanuit deze twee perspectieven. Uit de resultaten blijkt echter dat de rangschikking voor beide profielen dezelfde is en dus zal de volgorde eveneens behouden blijven bij het berekenen van het gemiddelde.

	Android/iOS	Cordova	Rhodes
Architect	59.99%	25.25%	14.75%
Ontwikkelaar	55.82%	30.77%	13.40%
Gemiddelde	57.90%	28.01%	14.08%

**Tabel 1. Overzicht van de totaalscores van de alternatieven.**

Tot slot worden de kosten en baten vergeleken. Er wordt geschat dat voor het ontwikkelen van Cordova applicaties ongeveer de helft van de tijd nodig is, vergeleken met native applicaties voor Android en iOS. Dit is het gevolg van de mogelijkheid om 100% van de applicatiecode te kunnen hergebruiken op Android en iOS. Er wordt tevens geschat dat het ontwikkelen van Rhodes applicaties iets langer duurt dan Cordova applicaties omdat er minder kant-en-klare componenten zijn die samen met Rhodes gebruikt kunnen worden. Deze bevindingen worden getoond in Figuur 3.

Uit deze figuur kan men concluderen dat Cord-



**Figuur 3. Visualisatie van de kosten-batenanalyse.**

ova en native Android/iOS een vergelijkbare kost-effectiviteit hebben. Voorlopig wordt het verlies in voordelen dus enkel gecompenseerd door een kortere projectduur. Indien echter binnenkort extra platformen (naast Android en iOS) ondersteund moeten worden, kan er drastisch op kosten bespaard worden door applicaties te ontwikkelen met Cordova op voorwaarde dat een zeker verlies van voordelen acceptabel is.

## 6. Gerelateerd werk

De methode die in dit onderzoek gehanteerd wordt om cross-platform tools te evalueren en te selecteren is gebaseerd op literatuurstudies van Jadhav en Sonar [30], [23]. Deze studies suggereren eveneens 3 methodes voor evaluatie waarvan het analytisch hiërarchisch proces van Saaty [27] gekozen werd.

Een andere bijdrage tot dit onderzoek is de studie van VisionMobile bij meer dan 2400 ontwikkelaars in 91 landen over het gebruik van cross-platform tools [19]. Deze studie onthult meerdere elementen die nuttig waren, zoals architecturen, alternatieven en redenen voor het selecteren of verlaten van bepaalde cross-platform tools.

## 7. Besluit

Dit artikel presenteert een methode voor het evalueren en selecteren van cross-platform tools voor gebruik in een bedrijfsomgeving en toont eveneens de resultaten na toepassing van deze methode op native Android en iOS en twee cross-platform tools, Apache Cordova en Motorola Rhodes. Het relatieve belang van de evaluatiecriteria werd vastgelegd door een ontwikkelaar en

een architect bij Capgemini en dus geeft dit onderzoek geen algemeen beeld van de volledige industrie maar eerder een beeld voor het geval van CapGemini, indien de bevroagde personen representatief zijn voor Capgemini in zijn geheel.

Het uiteindelijke resultaat van dit onderzoek is dat er momenteel nog weinig voordelen gehaald kunnen worden uit de beschouwde cross-platforms tools maar dit kan mogelijk veranderen in de toekomst, wanneer meer dan twee platformen ondersteund dienen te worden. Indien het verlies van voordelen acceptabel is, kan Cordova ervoor zorgen dat een hogere productiviteit en een lagere kost kan gerealiseerd worden.

## Referenties

- [1] Gartner, *Gartner Says Worldwide Smartphone Sales Grew 16 Per Cent in Second Quarter of 2008*, 2008. url: <http://www.gartner.com/it/page.jsp?id=754112> (bezoekt op 01-05-2013).
- [2] —, *Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 11.5 Per Cent Increase in Third Quarter of 2008*, 2008. url: <http://www.gartner.com/it/page.jsp?id=827912> (bezoekt op 01-05-2013).
- [3] —, *Gartner Says Worldwide Smartphone Sales Reached Its Lowest Growth Rate With 3.7 Per Cent Increase in Fourth Quarter of 2008*, 2009. url: <http://www.gartner.com/it/page.jsp?id=910112> (bezoekt op 01-05-2013).
- [4] —, *Gartner Says Worldwide Mobile Phone Sales Grew 17 Per Cent in First Quarter 2010*, 2010. url: <http://www.gartner.com/it/page.jsp?id=1372013> (bezoekt op 01-05-2013).
- [5] —, *Gartner Says Worldwide Mobile Device Sales Grew 13.8 Percent in Second Quarter of 2010, But Competition Drove Prices Down*, 2010. url: <http://www.gartner.com/it/page.jsp?id=1421013> (bezoekt op 01-05-2013).
- [6] —, *Gartner Says Worldwide Mobile Phone Sales Grew 35 Percent in Third Quarter 2010; Smartphone Sales Increased 96 Percent*, 2010. url: <http://www.gartner.com/it/page.jsp?id=1466313> (bezoekt op 01-05-2013).

- [7] —, *Gartner Says Worldwide Mobile Device Sales to End Users Reached 1.6 Billion Units in 2010; Smartphone Sales Grew 72 Percent in 2010*, 2011. url: <http://www.gartner.com/it/page.jsp?id=1543014> (bezocht op 01-05-2013).
- [8] —, *Gartner Says 428 Million Mobile Communication Devices Sold Worldwide in First Quarter 2011, a 19 Percent Increase Year-on-Year*, 2012. url: <http://www.gartner.com/it/page.jsp?id=1689814> (bezocht op 01-05-2013).
- [9] —, *Gartner Says Sales of Mobile Devices in Second Quarter of 2011 Grew 16.5 Percent Year-on-Year; Smartphone Sales Grew 74 Percent*, 2012. url: <http://www.gartner.com/it/page.jsp?id=1764714> (bezocht op 01-05-2013).
- [10] —, *Gartner Says Sales of Mobile Devices Grew 5.6 Percent in Third Quarter of 2011; Smartphone Sales Increased 42 Percent*, 2011. url: <http://www.gartner.com/it/page.jsp?id=1848514> (bezocht op 01-05-2013).
- [11] —, *Gartner Says Worldwide Smartphone Sales Soared in Fourth Quarter of 2011 With 47 Percent Growth*, 2012. url: <http://www.gartner.com/it/page.jsp?id=1924314> (bezocht op 01-05-2013).
- [12] —, *Gartner Says Worldwide Sales of Mobile Phones Declined 2 Percent in First Quarter of 2012; Previous Year-over-Year Decline Occurred in Second Quarter of 2009*, 2012. url: <http://www.gartner.com/it/page.jsp?id=2017015> (bezocht op 01-05-2013).
- [13] —, *Gartner Says Worldwide Sales of Mobile Phones Declined 2.3 Percent in Second Quarter of 2012*, 2012. url: <http://www.gartner.com/it/page.jsp?id=2120015> (bezocht op 01-05-2013).
- [14] —, *Gartner Says Worldwide Sales of Mobile Phones Declined 3 Percent in Third Quarter of 2012; Smartphone Sales Increased 47 Percent*, 2012. url: <http://www.gartner.com/newsroom/id/2237315> (bezocht op 01-05-2013).
- [15] —, *Gartner Says Worldwide Mobile Phone Sales Declined 1.7 Percent in 2012*, 2013. url: <http://www.gartner.com/newsroom/id/2335616> (bezocht op 01-05-2013).
- [16] —, *Gartner Says Asia/Pacific Led Worldwide Mobile Phone Sales to Growth in First Quarter of 2013*, 2013. url: <http://www.gartner.com/newsroom/id/2482816> (bezocht op 25-05-2013).
- [17] IDC, *IDC Raises Its Worldwide Tablet Forecast on Continued Strong Demand and Forthcoming New Product Launches*, 2012. url: <http://www.idc.com/getdoc.jsp?containerId=prUS23696912>.
- [18] „Developer Economics 2013”, VisionMobile, tech. rap., jan 2013, p. 1–60.
- [19] S. Jones, C. Voskoglou, M. Vakulenko, V. Measom, A. Constantinou en M. Kapetanakis, „Cross-platform developer tools 2012 Bridging the worlds of mobile apps and the web”, VisionMobile, tech. rap., 2012, p. 1–97. url: <http://www.visionmobile.com/product/cross-platform-developer-tools-2012/>.
- [20] P. Friese, *Cross platform mobile development*, 2012. url: <http://www.slideshare.net/peterfriese/cross-platform-mobile-development-11239246>.
- [21] M. Firtman, *Mobile HTML5*, 2012. url: <http://mobilehtml5.org/> (bezocht op 13-04-2013).
- [22] M. Mahemoff, *HTML5 vs native: the mobile app debate*, jun 2011. url: <http://www.html5rocks.com/en/mobile/nativedebate/> (bezocht op 27-02-2013).
- [23] A. S. Jadhav en R. M. Sonar, „Framework for evaluation and selection of the software packages: A hybrid knowledge based system approach”, *Journal of Systems and Software*, deel 84, nr. 8, p. 1394–1407, aug 2011, ISSN: 01641212. DOI: 10.1016/j.jss.2011.03.034. url: <http://linkinghub.elsevier.com/retrieve/pii/S016412121100077X>.
- [24] Research2guidance, *Cross-platform tools can save more than 30% of app development time*, 2012. url: <http://www.research2guidance.com/cross-platform-tools-can-save-more-than-30-of-app-development-time/> (bezocht op 01-06-2013).
- [25] Gartner, „Cross-Platform Mobile Development Frameworks”, tech. rap., 2011. url: <http://www.gartner.com/id=1848214>.

- [26] G. A. Miller, „The magical number seven, plus or minus two”, *The Psychological Review*, deel 63, nr. 1, p. 81–97, 1956. DOI: 10 . 1037 / h0043158.. url: <http://www.musanim.com/miller1956/>.
- [27] T. L. Saaty, *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*, reeks Advanced book program. McGraw-Hill, 1980, p. 287, ISBN: 9780070543713.
- [28] —, „How to make a decision: The analytic hierarchy process”, *European Journal of Operational Research*, deel 48, nr. 1, p. 9–26, sep 1990, ISSN: 03772217. DOI: 10 . 1016 / 0377 - 2217 (90 ) 90057 - I. url: <http://linkinghub.elsevier.com/retrieve/pii/037722179090057I>.
- [29] B. LeRoux, *PhoneGap Beliefs, Goals, and Philosophy*, 2012. url: <http://phonegap.com/2012/05/09/phonegap-beliefs-goals-and-philosophy/> (bezocht op 18-05-2013).
- [30] A. S. Jadhav en R. M. Sonar, „Evaluating and selecting software packages: A review”, *Information and Software Technology*, deel 51, nr. 3, p. 555–563, mrt 2009, ISSN: 09505849. DOI: 10.1016/j.infsof.2008.09.003. url: <http://linkinghub.elsevier.com/retrieve/pii/S0950584908001262>.