

## Een Agile software productline voor gameontwikkeling

*Erik Philippus*

*De tijd van de volledig mechanische eenarmige bandiet ligt al weer ver achter ons. Embedded software vormt tegenwoordig een essentieel onderdeel van een gokkast. Dit artikel schetst een beeld van de recente ontwikkelingen bij JVH Gaming Products op het gebied van softwareontwikkeling voor kansspelen.*



JVH Gaming is een Europees opererend bedrijf met als kernactiviteit het vermarkten en exploiteren van een brede scala aan kansspelen in horecabedrijven, casino's en amusementscentra. De businessunit JVH Gaming Products in Tilburg is verantwoordelijk voor de productie van deze spelen. Een voorbeeld is het traditionele rollenspel Simply Wild, waarvan JVH er in Nederland thans meer dan drieduizend exploiteert.

De markt voor kansspelen is in de meeste Europese landen strikt gereguleerd om fraude en excessieve goklust te voorkomen. Vanwege verschillen in lokale cultuur en wetgeving is het voor JVH niet mogelijk een en hetzelfde product te maken voor heel Europa. Dit heeft een duidelijk stempel gedrukt op de inrichting en werkwijze van de R&D-afdeling. In de loop der tijd ontstonden er diverse kleine, multidisciplinaire teams voor specifieke landen of regio's. Eén zo'n team maakte dan vaak een variant van een kansspel dat al eerder was ontwikkeld voor een ander land.



De beproefde werkwijze bestond eruit de bestaande software over te nemen, daar alle landspecifieke wijzigingen in door te voeren en vervolgens een nieuw, zelfstandig product uit te brengen. Om zo snel mogelijk nieuwe spellen in de hectische markt te kunnen zetten, implementeerden de teams gemeenschappelijke, landonafhankelijke functionaliteit vaak op verschillende manieren. In de waan van de dag beschouwden ze investeringen in kwaliteitsattributen als herbruikbaarheid en uitbreidbaarheid al snel als een sluitpost.

Voor de relatief simpele, traditionele kansspelen was dit nog wel te doen. Met name de opkomst van videotechnieken in de kansspelindustrie leidde echter tot een aanzienlijke groei in omvang en complexiteit van de embedded software. De gestaag toenemende variabiliteit in de waaier van producten begon toen z'n tol te eisen. Het viel JVH steeds zwaarder om snel te reageren op nieuwe trends in de markt, technologische innovaties en continu veranderende wetgeving. Vandaar dat het bedrijf op zoek is gegaan naar manieren om zijn portfolio van kansspelen voor de Europese markt efficiënter te ontwikkelen en te onderhouden.

### **Platform releaseschema**

Een kleine twee jaar geleden heeft JVH besloten over te stappen op de Agile-werkwijze. Deze nieuwe aanpak hebben we eerst geïntroduceerd op de R&D-afdeling, waar we de kleine landenteams hebben samengevoegd en omgevormd tot grotere, multidisciplinaire teams werkend met Scrum als Agile-projectmanagementmethode. In een tweede fase hebben we de overige afdelingen zoals verkoop en marketing in lijn gebracht. De omschakeling van eenmalige, landspecifieke producten maken naar werken aan een portfolio van gerelateerde producten bestemd voor een aantal landen heeft uiteraard wel wat voeten in de aarde gehad.

Pas toen de gehele R&D-afdeling over was op de Scrum-werkwijze werd duidelijk dat marketing- en salesmanagers een bruikbaar instrument in handen hadden gekregen om te komen tot een strategische releaseplanning. In toenemende mate werd vooraf zichtbaar welke producten in welke landen zouden verschijnen. Bij de samenstelling van de product backlogs kon dan ook steeds meer rekening houden met herbruikbare softwareonderdelen.

De ontwikkeling van een softwarebibliotheek met generieke functionaliteit leek hierdoor eindelijk vaste grond onder de voeten te krijgen. Soms zijn oude patronen echter weerbarstig, en in de hectiek van alledag vergeten we nogal eens de horizon verder te leggen dan het eerstvolgende product. Ook bleek in de praktijk dat een product backlog bestaande uit een combinatie van productspecifieke en generieke features lastig te hanteren kan zijn. De eigenaar van de softwarebibliotheek voelde zich vaak als een tweede kapitein op het schip. Dit heeft geleid tot de vorming van een afzonderlijk Scrum team voor het JVH-softwareplatform.

De ervaring leert dat werkzaamheden voor een softwareplatform al snel naar het tweede plan schuiven als ze moeten concurreren met urgente activiteiten voor de lopende projecten. Het platformteam heeft daarom een vaste bemanning gekregen die zonder al te veel onderbrekingen de productie van herbruikbare softwarecomponenten ter hand kan nemen aan de hand van een eigen product backlog. In het eveneens geformeerde 'Scrum of Scrums-team' spreken alle belanghebbenden af wanneer er generieke functionaliteit beschikbaar komt in het platform. Deze benadering heeft bij de producteigenaren het belang onderstreept om vooraf met elkaar na te denken over het complete productportfolio.

Om de zichtbaarheid van het werk aan het softwareplatform te verhogen, hebben we het platform releaseschema ingevoerd, in vereenvoudigde vorm weergegeven in Figuur 1. Kenmerkend is het vaste ritme waarmee versies van het platform beschikbaar komen, in dit geval drie Scrum-sprints (iteraties), overeenkomend met een tijdvak van zes weken. In nauw overleg met alle 'afnemers' van het platform stelt het platformteam een backlog samen van generieke features, die het regelmatig actualiseert. De relatieve omvang van iedere te realiseren feature wordt vooraf ingeschat en uitgedrukt in *story points*.

	Feature A	Feature B	Feature C	Feature D	Feature E	Feature F	Feature G	Feature H	Feature I	Feature J
Effort Estimation (story points)	6	4	8	14	6	5	9	16	11	12
Remaining Work	0	2	2	5	1	0	1	7	7	12
Platform Release 1	●					●				
Iteration 1.1	3		4			4		3		
Iteration 1.2	2			4		1		6		
Iteration 1.3	1			5	3		5			
Platform Release 2		●	●		●		●			
Iteration 2.1		2	2		2		3		4	
Iteration 2.2		2	2		1		3		5	
Iteration 2.3				2			1	2		9

Figuur 1: Het platform releaseschema laat in één oogopslag zien wat de huidige status is van het softwareplatform en wanneer nieuwe features beschikbaar komen.

Een story point is een arbitraire maat die het Scrum-team gebruikt om de inspanning te kwantificeren die het zal kosten om een feature (een *user story* in Scrum-termen) te implementeren. Na enige tijd wordt duidelijk hoeveel story points het Scrum-team bij benadering in één sprint kan realiseren, de zogeheten *team velocity*. Deze waarde gebruikt het team dan om de features op een realistische wijze te verdelen over de verschillende sprints.

Het schema laat in één oogopslag zien wat de huidige status is van het softwareplatform en wanneer nieuwe features beschikbaar komen. Marketing- en salesmanagers hebben hiermee een basis om strategische plannen op te stellen. Ook kunnen ze de juiste afwegingen maken indien zich in de markt onverwachte wijzigingen voordoen die vragen om nieuwe features of indien reeds ingeplande features eerder beschikbaar zijn. In principe is het mogelijk om na afloop van elke sprint de verdeling van de features over de komende sprints bij te stellen en daarmee de inhoud van de volgende platformrelease af te stemmen op de actuele marktvrage. Niet alleen verbetert dit de efficiency, door hergebruik van softwaremodules, maar het zorgt ook voor een aanzienlijke grotere slagvaardigheid voor de JVH organisatie als geheel.

## **Kanban**

In de praktijk bleek de Scrum-werkwijze niet optimaal te passen bij het platformteam. Met het oog op de diversiteit van de te realiseren features hadden we dat team samengesteld uit softwareontwikkelaars met elk specifieke domeinkennis en exclusieve competenties. Dit bemoeilijkte de vorming van een hecht, multifunctioneel team. Pogingen om bij aanvang van elke iteratie een gemeenschappelijk sprintdoel af te spreken, ervoeren de betrokkenen na enige tijd als enigszins geforceerd.

Daarnaast willen de afnemers van het platform zich op elk moment een juist beeld kunnen vormen van de status van elke feature. Volgens de Scrum-methodiek delen we een feature op in taken die we op het Scrum-planningbord kunnen aangeven als 'to be done', 'in progress' of 'done'. Hieruit is echter niet direct de actuele status van de feature als geheel te destilleren.

Deze problematiek heeft geleid tot een volgende verbeterstap in de Agile-werkwijze binnen het platformteam. Deze hebben we gevonden in de combinatie van Scrum met Kanban. Kanban, een term die bestaat uit de Japanse woorden voor 'zichtbaar' (*kan*) en 'kaart' (*ban*), vormt de basis van Lean Manufacturing en is erop gericht de productiestroom te optimaliseren en zichtbaar te maken.

Net als Scrum maakt de aanpak gebruik van een bord dat de actuele status van de werkzaamheden weergeeft. Bij Kanban is de focus echter meer gericht op de feature als geheel dan op de status van de afzonderlijke taken. Op het bord vinden we voor elke feature een kaart, die gedurende realisatie van links naar rechts beweegt. Het verschil met Scrum is dat Kanban de fase 'in progress' onderverdeelt in meerdere stadia, waaronder 'elaboration', 'production', 'integration', 'test' en 'deployment'. Hierdoor is direct zichtbaar in welk stadium een feature verkeert.

Bijkomend voordeel van Kanban is dat het een bovengrens stelt aan het aantal features dat zich in één stadium mag bevinden. Deze beperking van het werk 'in uitvoering' zorgt voor een goede doorstroming. Kanban voorkomt hiermee dat activiteiten zich ophopen, met negatieve gevolgen voor de efficiency en doorlooptijd van de features.

Op deze wijze kan het platformteam op het juiste moment de door de spelontwikkelaars benodigde features leveren. Deze features zijn ontworpen met herbruikbaarheid als voornaam kenmerk, waardoor de ontwikkelteams hetzelfde implementatietraject volgen, in ieder geval voor de gemeenschappelijke (landonafhankelijke) functionaliteit.

Op deze wijze is JVH Gaming Products in vrij korte tijd omgeschakeld van relatief afzonderlijk opererende landenteams naar een Agile software productlijn voor zijn portfolio van kansspelen.