31-12-2011

****

|  |
| --- |
| Advies participatie van bezoekers bij toekomstige events op Hyves | Jonathan de Zoete |

|  |  |
| --- | --- |
| HYVES VOORSPELLEN | Afstudeerverslag |

**Studentnummer**20022095

**Docenten**  
C. Heydra  
D. Mast

**Opdrachtgever**  
D. Vismans

**Bedrijf**  
Hyves B.V.

# Referaat

Jonathan de Zoete, procesverslag over het tot stand komen van een advies over participatie van bezoekers bij toekomstige events. Hiervoor wordt een casus gebruikt, namelijk: Hyves Voorspellen. Hyves.nl, Amsterdam, januari 2012.

Student Jonathan de Zoete heeft in maart tot en met augustus 2011 zijn afstudeeropdracht uitgevoerd bij Hyves.nl. Dit verslag zal inzicht geven over de achtergrond van het project en de handelswijze van de student.

Descriptoren:

* IAD
* User Interface Design
* Database ontwerp
* Programmeren
* Controlegroep gebruikerstesten
* Online gaming
* Adviesrapport

# Voorwoord

In maart tot en met augustus 2011 heb ik 25 weken aan het project Hyves Voorspellen mogen werken. Dit project is de afsluiting van mijn deeltijd HBO studie Communication & Multimedia Design aan de Haagse Hogeschool. Dit verslag is een reflectie op de wijze waarop ik het project heb doorlopen. Het bevat de stappen die ik heb genomen en de keuzes die ik heb gemaakt.

Hyves Voorspellen is de werktitel van het project en de casus die ik heb gebruikt om een vraag met een grotere reikwijdte te kunnen beantwoorden: “Is het mogelijk het onderwerp van de conversatie tussen gebruikers te sturen naar een gebeurtenis in de toekomst?”. Daar komt ook de officiële titel van het project vandaan:

Advies participatie van bezoekers bij toekomstige events op Hyves.

In dit voorwoord wil ik Hyves, en in het bijzonder dhr. Vismans bedanken voor het vertrouwen dat ik heb gekregen om dit project, dat voor Hyves een belangrijk onderdeel is geweest, naar eigen inzicht in te richten en te ontwikkelen.  
Hiernaast wil ik dhr. Coumans bedanken voor zijn rol als scriptiebegeleider op Hyves.  
  
Veel leesplezier, en vergeet vooral niet na afloop naar voorspellen.nl te gaan!

Jonathan de Zoete

Den Haag, 31-12-2011

Contents

[Referaat 2](#_Toc313887039)

[Voorwoord 3](#_Toc313887040)

[Inleiding 6](#_Toc313887041)

[1. Deel I 7](#_Toc313887042)

[1.1 Bedrijf 7](#_Toc313887043)

[1.2 Stakeholders 7](#_Toc313887044)

[1.3 Probleemstelling 7](#_Toc313887045)

[1.4 Doelstelling 8](#_Toc313887046)

[1.5 Opdracht 8](#_Toc313887047)

[1.6 Systeemeisen 9](#_Toc313887048)

[1.7 Methode 9](#_Toc313887049)

[1.8 Strategie 11](#_Toc313887050)

[1.9 Planning 14](#_Toc313887051)

[Fase 1: Definitiestudie 14](#_Toc313887052)

[Fase 2: Pilotontwikkeling 15](#_Toc313887053)

[Fase 3: Invoering 15](#_Toc313887054)

[2. Deel II 16](#_Toc313887055)

[Fase 1: Definitiestudie 16](#_Toc313887056)

[Activiteit 1: Stel plan van aanpak op 16](#_Toc313887057)

[Activiteit 2: Definieer ontwikkelscenario 17](#_Toc313887058)

[Activiteit 3: Definieer systeemeisen 20](#_Toc313887059)

[Activiteit 4: Bepaal systeemconcept 21](#_Toc313887060)

[Activiteit 5: Beschouw technische structuur 27](#_Toc313887061)

[Activiteit 6: Beschouw organisatorische inrichting 29](#_Toc313887062)

[Activiteit 7: Stel pilotplan op 29](#_Toc313887063)

[Fase 2: Pilotontwikkeling 31](#_Toc313887064)

[Pilot 1: Database, deel 1 31](#_Toc313887065)

[Pilot 2: Front end, deel 1 37](#_Toc313887066)

[Pilot 2: Back end, deel 1 47](#_Toc313887067)

[Invoering voor testgroep 52](#_Toc313887068)

[Pilot 1,2 en 3: deel 2 (bugfixes) 54](#_Toc313887069)

[Invoering 54](#_Toc313887070)

[Definitiestudie 2 55](#_Toc313887071)

[Pilot 2 en 3: deel 3 55](#_Toc313887072)

[Invoering 57](#_Toc313887073)

[Definitiestudie 3 57](#_Toc313887074)

[Pilot 2 en 3: deel 4 57](#_Toc313887075)

[Invoering 58](#_Toc313887076)

[Definitiestudie 4 59](#_Toc313887077)

[Pilot 2 en 3: deel 5 59](#_Toc313887078)

[Fase 3: Invoering 61](#_Toc313887079)

[Activiteit 2: Stel meetplan op 62](#_Toc313887080)

[Activiteit 3: Voer pilots in 62](#_Toc313887081)

[Activiteit 4: Draag pilots over 62](#_Toc313887082)

[Activiteit 5: Evalueer testresultaten 62](#_Toc313887083)

[3. Deel III, conclusie 65](#_Toc313887084)

[3.1 Procesevaluatie 65](#_Toc313887085)

[3.1.1 Definitiestudie 65](#_Toc313887086)

[3.1.2 Pilotontwikkeling 67](#_Toc313887087)

[3.1.4 Invoering 70](#_Toc313887088)

[3.1.5 Strategie 70](#_Toc313887089)

[3.1.6 Methode 70](#_Toc313887090)

[3.2 Productevaluatie 71](#_Toc313887091)

[3.2.1 opdracht 71](#_Toc313887092)

[3.2.2 Doelstelling 71](#_Toc313887093)

[3.2.3 Probleemstelling 71](#_Toc313887094)

Literatuurlijst 72

[Verklarende woordenlijst 73](#_Toc313887095)

Bijlagen

A Afstudeerplan  
B Plan van aanpak  
C GANTT planning  
D Scenarioplan  
E Taakdiagrammen  
F Klassediagram  
G Adviesrapport

# Inleiding

Hyves is een Nederlands sociaal netwerk met een gebruikersgroep van meer dan 9 miljoen personen, jong en oud. Bij een sociaal netwerk draait het om het verbinden van de gebruikers en dat uit zich in de slagzin van Hyves: “Always in touch with your friends”.

Hyves is opgericht in 2004 en is sindsdien uitgegroeid tot een van de meest bezochte websites in Nederland met gemiddeld 150.000.000 pageviews per dag.

Hyves is na de oprichting al snel marktleider in Nederland geworden in het segment sociale netwerken. De opmars van met name Facebook zet die positie in Nederland sterk onder druk en het lijkt slechts een kwestie van tijd voordat Facebook in Nederland meer gebruik heeft dan Hyves. Gebruik wordt gemeten door het aantal gebruikers en de tijd die zij spenderen op de website.

Hyves wil zich uiteraard niet vrijwillig naar de slachtbank laten leiden en zoekt nieuwe manieren om de marktleider te blijven. Een van deze nieuwe manieren is het proberen te sturen van conversaties. In plaats van puur een communicatiemiddel te zijn, ook een redactioneel aspect aanbrengen.  
Op Hyves wordt er veel over voetbal gepraat, sterker nog, het is het meest populaire gespreksonderwerp op Hyves. Er is bijvoorbeeld gemeten dat na bepaalde voetbalwedstrijden meer dan 10% van alle berichten op Hyves over voetbal ging. Dat gebeurt echter vooral na een wedstrijd.

Hyves wil weten of het mogelijk is om de conversatie zo te sturen dat er van tevoren ook over een voetbalwedstrijd wordt gepraat. Om de gebruikers uit te dagen om erover te praten is het idee ontstaan om een voetbalpoule op Hyves te ontwikkelen: Hyves Voorspellen.

Deze afstudeeropdracht bestaat uit het ontwikkelen van die voetbalpoule. Aan de hand hiervan moet er een advies worden gegeven of het mogelijk is daadwerkelijk mogelijk is de conversatie te sturen.

Er kan natuurlijk alleen goed antwoord op de vraag worden gegeven als het product van een gedegen kwaliteit is. Een goed idee met een slecht product is immers waardeloos, omdat mensen het niet zullen gebruiken. Dit verslag is een verantwoording van het ontwikkelen van het product. Zowel het creatieve gedeelte, als het technische gedeelte komen aan bod.

Het verslag bestaat uit drie delen. Het eerste Deel gaat over de start van de opdracht. Waarom wordt het gedaan en hoe gaat de student het uitvoeren. Het tweede deel gaat over de uitvoering van de opdracht. Stap voor stap wordt beschreven hoe de student het project is doorlopen. Het derde deel is een proces- en productevaluatie.

# 1. Deel I

## 1.1 Bedrijf

Hyves is een organisatie met ongeveer 165 werknemers. Het is verdeeld in verschillende afdelingen. De verschillende afdelingen zijn Sales, Marketing, Community, QA, HR, Product, System Engineering en Development.

De afdeling Development zorgt voor de ontwikkeling en onderhoud van de software van Hyves. Dat is zonder de afdeling system engineering, die voor de infrastructuur zorgt (serverpark, verbindingen etc.). Met ongeveer 30 ontwikkelaars beslaat het zo’n 20 % van het totaal aantal werknemers. Vanwege het hoge aantal buitenlandse werknemers op Development, ongeveer 50%, is de voertaal Engels.

De taakverdeling is op de afdeling Development opgesplitst in verschillende lagen. Kort gezegd, worden op directieniveau strategische keuzes gemaakt en plannen bedacht. Deze worden vervolgens met nauw overleg met de head of development verdeeld over bepaalde teams.

Een team bestaat uit een teamleider die enkele medewerkers aanstuurt. Ook worden er één of meerdere product managers aan een project gecommitteerd. De teams hebben elk hun eigen verantwoordelijkheden, maar er wordt ook veel samengewerkt.  
Ik werk bij Hyves in het team Sales & Business Intelligence. Dit team zorgt voor ontwikkeling op de website die met geld te maken heeft. De activiteiten beslaan bijvoorbeeld aan de ene kant de techniek voor het inboeken en tonen van advertenties, maar aan de andere kant ook het interface design voor statistiekenpagina’s van aan bedrijven verkochte ‘branded hyves’.

Tijdens mijn afstudeerproject zal ik vooral samenwerken met het design team. Mijn begeleider, Jeroen Coumans, is teamleider van dit designteam en is usability designer binnen Hyves.

## 1.2 Stakeholders

De volgende personen zijn belangrijk binnen dit project en kunnen genoemd worden in het verslag:

Vanuit Hyves:

- MSc David Vismans, CTO. David is de opdrachtgever.  
 - MCMD Jeroen Coumans, Lead Designer. Jeroen is mijn begeleider.  
 - Hanno ten Hoor, product designer. Hanno helpt bij het ontwikkelen van de UI.  
 - BIT Kilian Marjew, Software Engineer. Kilian ondersteunt bij het programmeren.

Vanuit de Haagse Hogeschool:

- MSc Chris Heydra. Chris is mijn scriptiebegeleider.  
 - MSc Danica Mast. Danica is expert begeleider.

## 1.3 Probleemstelling

Hyves heeft er strategisch voor gekozen om een lokaal platform te zijn en dat betekent dat het zich puur richt op de Nederlandse markt. In Nederland is het niet zo simpel om het ledenaantal te vergroten. De reden daarvoor is dat vrijwel iedereen kent Hyves en is er lid van, of kiest er bewust voor om dat niet te zijn, om welke reden dan ook. En dat is de essentie van het probleem. Hyves is groot geworden door een snelle groei in het aantal leden en bedrijven zoeken uiteraard altijd naar groei.

De probleemstelling is dan ook als volgt:

**Hyves is op zoek naar groei, maar kan dit niet doen door nieuwe leden aan te trekken.**

Dat wetende, moet de groei ergens anders gezocht worden.

Als je geen nieuwe leden kunt aantrekken, is het logisch om te proberen bestaande gebruikers sterker aan je binden en het aantal ‘minutes spent’ te verhogen.

Communicatie op Hyves gebeurt vooral over wat er in het verleden is gebeurd, of wat nu gebeurt. Toekomstige gebeurtenissen worden niet of nauwelijks besproken en dat kan misschien anders.

Hyves wil de communicatie gaan sturen zodat er ook over toekomstige gebeurtenissen of events wordt gepraat. Daaruit vloeit de doelstelling voort.

## 1.4 Doelstelling

Door te kijken naar wat Hyvers zoal bezighoudt is gebleken dat er erg veel over voetbal wordt gepraat, er is gemeten dat in sommige weken 10% van de WieWatWaars (persoonlijke berichten) over voetbal ging. Uiteraard wordt op deze cijfers teruggekomen in het uit te voeren onderzoek.

In het idee om de communicatie te sturen naar toekomstige events, is het populairste gespreksonderwerp op Hyves een geschikte casus om te zien of dit kan werken. Dat heeft het idee Hyves Voorspellen opgeleverd:

Hyves Voorspellen is het voorspellen van voetbalwedstrijden op Hyves en de communicatie daaromheen.

De doelstelling heeft te maken met de casus die gebruikt wordt om antwoord te kunnen geven op de probleemstelling:

**Adviseren over de vraag of het mogelijk is communicatie te sturen naar events in de toekomst, met het ontwikkelen van het Hyves Voorspellen platform als casus.**

Om een richting aan te houden in het onderzoeken van het advies, is de volgende onderzoeksvraag geformuleerd:

**Kan de participatie van gebruikers op Hyves worden verhoogd door de communicatie te sturen richting gebeurtenissen in de toekomst?**

## 1.5 Opdracht

De casus moet laten zien of het mogelijk is om de communicatie van gebruikers te sturen. Mocht het onderzoek aangeven dat er groei in de diepte mogelijk is, dan is dit een signaal dat de goede richting in is geslagen. Groei in de diepte betekent het verhogen van het aantal minuten dat iemand op Hyves spendeert. Omdat Hyves wil dat bestaande gebruikers Hyves meer gaan gebruiker, zijn deze gebruikers ook direct de doelgroep.   
Na het zien van de effecten van deze casus kan Hyves verdere stappen ondernemen en soortgelijke producten ontwikkelen, mochten ze dat willen.

Om te kunnen zien of Hyves voorspellen resultaat boekt, zal het gebruik van Hyves Voorspellen uitgebreid gemeten worden.

Als resultaat is er enerzijds het werkende Hyves Voorspellen als product, en anderzijds het adviesrapport.

## 1.6 Systeemeisen

Na onderzoek, dat later in het verslag beschreven zal worden, heb ik de volgende basissysteemeisen opgesteld:

1. Hyves voorspellen manifesteert zich op 3 plekken binnen Hyves:
   1. Een overzichtspage/homepage, waar vandaan naar alle wedstrijden kan worden genavigeerd.
   2. Een itempagina, waar alle acties rondom het event te doen zijn.
   3. De homepage, waar gebruikers die meedoen proactief worden uitgenodigd en uitgedaagd om te voorspellen. Ook updates van voorspellingen zijn te zien op de homepage (`buzz-items`).
2. Op de homepage moet een `blok` zichtbaar worden, enkele dagen voor een bepaalde wedstrijd, waar die wedstrijd direct is te voorspellen. Ook moet te zien zijn of vrienden deze wedstrijd al voorspeld hebben.  
   Na de wedstrijd moet dit blok tonen wat de uitslag is en welke vrienden dat goed hadden.
3. Als een voorspelling gedaan wordt, moet dit als update verspreid worden naar de homepage van vrienden van diegene (buzz-item).
4. De Hyves Voorspellen homepage moet buiten de te voorspellen wedstrijden weergeven, ook laten zien wat de status is van het systeem. Dit houdt in dat te zien is hoeveel punten of andere beloningen diegene heeft verzameld en hoe goed hij is in vergelijking met vrienden.
5. De item pagina moet een pagina worden die volledig op dat event of wedstrijd is toegespitst.
   1. De voorspelling moet daar te doen zijn, alswel als kunnen worden aangepast.
   2. Er moet te zien zijn wat anderen voorspellen voor deze wedstrijd.
   3. Er moet kunnen worden gepraat over de wedstrijd.
   4. Na de wedstrijd moet de uitslag van de wedstrijd en beloningen te zien zijn

In Deel 2, activiteit 3 van dit verslag wordt het tot stand komen van de systeemeisen beschreven.  
Deze systeemeisen worden uiteraard specifieker naarmate het project is verder is.

## 1.7 Methode

Om het de projectontwikkeling structuur te geven en een leidraad te hebben, is er een ontwikkelmethode nodig. Een leraar had mij attent gemaakt op de RAD methode, Rapid Application Development. RAD is echter meer toegespitst op de ontwikkeling met een groep experts, terwijl ik de casus grotendeels zelf doorloop. Ook staan de systeemeisen bij RAD vast van begin tot eind, wat niet het geval is in deze casus. Daarom is de RAD methode niet geschikt voor dit project.  
 In mijn onderzoek naar RAD kwam ik meerdere ander ontwikkelmethodes tegen, waaronder RUP en IAD. RUP staat voor Rational Unified Process, IAD staat voor Iterative Application Development.

RUP erkent de problemen die er zijn met projecten, zoals veranderende requirements. Ik denk dat dit project ook veranderende requirements zal hebben, aangezien de oplossing van het probleem waarvoor ik dit project voor doe niet van tevoren helemaal te specificeren is. Dat spreekt mij dus aan aan RUP. De oplossing die RUP voor dit probleem heeft spreekt mij echter iets minder aan. Het project dient opgedeeld te worden in bepaalde delen die apart worden opgeleverd. Voor die delen moet de hele methode opnieuw worden herhaald. Dit levert voor een kleiner project veel dubbele documenten op. Het zou beter zijn om bepaalde onderdelen te kunnen itereren, zoals dat bij de IAD methode het geval is.

Het iteratieve karakter van IAD lijkt ook goed aan te sluiten bij de ontwikkelcyclus op Hyves. Die ontwikkelcyclus bestaat kort gezegd uit een wekelijkse code release. Een iteratie op Hyves bestaat dan ook uit 1 of meerdere weken.   
Er is gekozen voor deze ontwikkelcyclus omdat dit ideaal is voor een website waarbij snel ingesprongen moet worden op de veranderende wensen van zijn gebruikers. IAD is ook in dit aspect geschikt, aangezien het uitgaat van veel interactie tussen de ontwikkelaars en de gebruikers, waarbij de systeemeisen aan het begin niet vast staan.  
  
IAD heeft verschillende ontwikkelstrategieën en staat toe om daar flexibel mee om te gaan. Hierdoor is het goed toe te passen op het type applicatie dat wordt ontwikkeld. De strategie wordt in de volgende paragraaf beschreven.   
  
De I uit IAD wordt ook wel eens voor Incrementeel. IAD kan goed gebruikt worden om een product of prototype op te leveren dat incrementeel uitgebreid kan worden. Ook dit past uitstekend bij Hyves. Van elk product wordt uitgegaan van bepaalde systeemeisen. Als er echter een idee voor een uitbreiding of extra functionaliteit ontstaat, en dit de als de moeite waard wordt gezien om te ontwikkelen, zal dat ook gedaan worden in een extra iteratie. De grenzen tussen nieuwbouw en onderhoud vervagen hier.

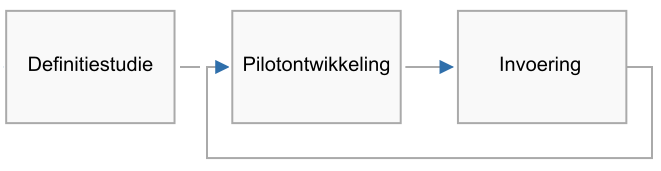
De IAD methode is dus geschikt omdat:

- Het aansluit bij de ontwikkelcyclus op Hyves  
- Het gaat uit van veel interactie tussen ontwikkelaar en gebruiker  
 - De strategie kan worden toegespitst op het type applicatie  
 - Het incrementeel karakter goed bij het product past

## 

## 1.8 Strategie

Bij de IAD methode hoort een strategie, de IAD methode is namelijk uit te voeren met meerdere strategieën. Belangrijke verschillen in strategieën zijn het al dan niet vaststaan van de systeemeisen, het invoeren in 1 keer of geleidelijk. De verschillende strategieën van IAD zijn: i*ncrementeel opleveren*, big-bang-invoeren, *Incrementeel opleveren*, *evolutionair ontwikkelen,* en hybride ontwikkelen, wat een combinatie van eerder genoemde strategieën is.

*Incrementeel opleveren* schrijft voor dat alle systeemeisen van tevoren vastgesteld zijn, waarna deze incrementeel ontwikkeld en ingevoerd worden in het bedrijf. Aangezien ons product geen vaste systeemeisen heeft, valt deze strategie af.  
  
  
**Inzet: iteratie fasen bij incrementeel opleveren**  
Bij *big-bang-invoeren* komen zowel het systeemconcept als de systeemeisen iteratief tot stand. Dit lijkt al meer op de strategie die bij mijn project past. Deze strategie schrijft echter voor dat het product in 1 keer ingevoerd wordt. Ik wil echter na invoering nog door ontwikkelen, waardoor big-bang-invoeren afvalt.  
Incrementeel opleveren, dat erg veel lijkt op RAD, heeft ook als eis dat het systeemconcept en –eisen volledig zijn gespecificeerd aan het begin. De strategie *incrementeel opleveren* valt hierdoor ook af.  
Bij evolutionair ontwikkelen worden de drie fases van IAD – Definitiestudie, pilotontwikkeling en invoering – bij elke iteratie doorlopen. De systeemeisen en het systeemconcept worden dus na elke iteratie herzien.

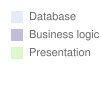
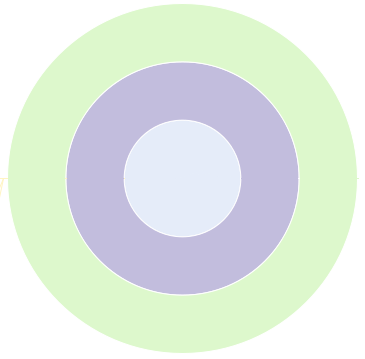
  
**Inzet: iteratie fasen bij evolutionair ontwikkelen**

Evolutionair ontwikkelen sluit dus het best aan bij dit project. Er is echter een aspect waar ik vanaf wil wijken.   
Gezien er bij Hyves elke week nieuwe functionaliteit online wordt gezet, zullen de iteraties na de eerste invoering een week duren. Het kan bijvoorbeeld zijn dat nog niet alle functionaliteit in de definitiestudie klaar is. In dat geval wordt de volgende pilot zonder definitiestudie uitgevoerd, zoals bij incrementeel opleveren het geval is.  
  
De strategie die ik ga gebruiken is dus ‘hybride ontwikkelen’, waarbij ik vooral uitga van de strategie *evolutionair ontwikkelen*. Dit in combinatie met een eventuele iteratiestrategie van *incrementeel opleveren*.

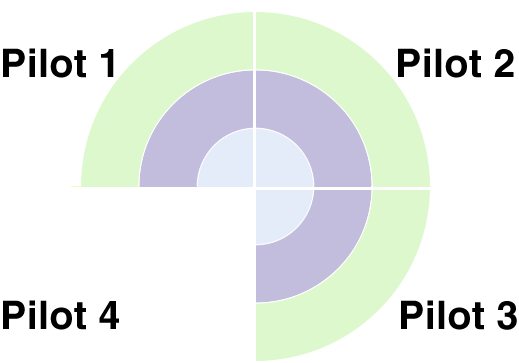
Deze strategie is van toepassing op elke pilot. In het afstudeerplan heb ik drie pilots beschreven:

* Pilot 1: Database
* Pilot 2: Design
* Pilot 3: Bouw back end

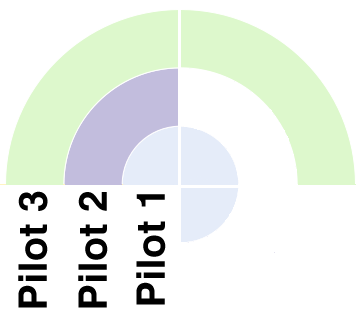
Deze 3 pilots staan gelijk aan de drie delen van het 3-tier model. Het 3-tier model beschrijft verschillende delen van een systeem als schillen van een software systeem. Het voordeel hiervan is dat de technische werking zo goed mogelijk gescheiden blijft, en er aanpassingen gedaan kunnen worden in een bepaalde schil, terwijl de andere schillen intact gelaten worden.

  
**Inzet: 3-tier model**

Bij het ontwikkelen van applicaties, bijvoorbeeld met de IAD methode, bestaan pilots vaak uit een taartpunt van deze cirkel. Alle taartpunten bij elkaar vormen dan het gehele systeem.

  
**Inzet: Voortgang bij pilots als taartpunten. 3 van de 4 pilots zijn klaar**

Ik kies voor een ander systeem. Door de pilots niet te benoemen per taartpunt, maar per schil, worden de pilots extra dynamisch. Als je een taartpunt uit het 3-tier systeem neemt, werk je steeds aan alle 3 de schillen tegelijk. Nu kan ik dat nog steeds doen, volgens IAD is het namelijk geen probleem om pilots tegelijkertijd te ontwikkelen. Maar als ik wil, kan ik nuk nu ook alleen aan één schil tegelijk werken. Het volgende figuur illustreert dit.

  
**Inzet: Voortgang bij pilots als schillen. Van drie pilots zijn een verschillend aantal delen klaar.**

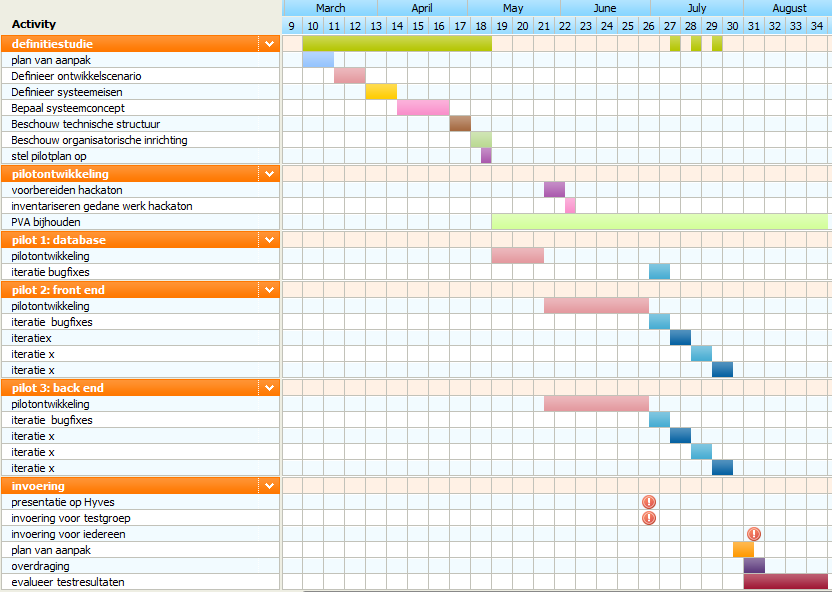
In een eerste iteratie zal ik de de pilot uitwerken vanuit de initiele systeemeisen. De daarop volgende pilots zijn iteraties van 1 week, waarbij feedback de belangrijkste input is voor bijgestelde systeemeisen. Van tevoren weet ik dus niet wat voor werk er in deze iteraties gedaan zal worden. Daarom neem ik ze in mijn planning op als iteratie x.  
Ik kies voor iteraties van één week, omdat Hyves met deze ontwikkelcyclus werkt. Eén maal per week, op donderdag, worden aanpassingen aan de website *live gezet*. Op deze manier kan er snel worden ingesprongen op wensen van gebruikers.  
Dit principe is een paradigma in softwareontwikkeling waarbij het belangrijk is een product snel op de markt te brengen en te laten evolueren om zo snel mogelijk aan de wensen van de gebruikers te voldoen. Het wordt ‘Release Early, Release Often’ (RERO) genoemd, en werd door Eric S. Raymond geïntroduceerd.

  
- Eric S. Raymond

## 1.9 Planning

De IAD methode heeft drie hoofdfases: de definitiestudie, de pilotontwikkeling en de invoering. In het nu volgende deel van het verslag zal ik beschrijven hoe de planning tot stand is gekomen.

**Inzet: GANTT planning**



Fases bij de IAD methode zijn een afgeleide van de ontwikkelstrategie en bevatten meerdere activiteiten. Deze activiteiten zijn gericht op het opleveren van een of meerdere producten. IAD schrijft de volgende fasen en activiteiten voor:

### Fase 1: Definitiestudie

Activiteit 1: Stel plan van aanpak op  
Activiteit 2: Definieer ontwikkelscenario  
Activiteit 3: Definieer systeemeisen  
Activiteit 4: Bepaal systeemconcept  
Activiteit 5: Beschouw technische structuur  
Activiteit 6: Beschouw organisatorische inrichting  
  
  
  
In het boek ‘Het evolutionair ontwikkelen van informatiesystemen’ van R.H.J. Tolido staan de activiteiten uitgebreid beschreven. Aan de hand van deze beschrijving schat ik in hoeveel tijd ik ongeveer per activiteit nodig heb en neem dit op in een GANTT chart.

### Fase 2: Pilotontwikkeling

Activiteit 1: stel plan van aanpak op  
Activiteit 2: Ontwerp software bouweenheden   
Activiteit 3: Bouw software bouweenheden   
Activiteit 4: Beoordeel en test pilotdeel

Activiteit 1, het PVA is een activiteit die door de hele fase heen loopt. Het PVA moet gedurende de gehele fase worden geactualiseerd met aanpassingen en toevoegingen. Activiteit 2 t/m 4 zullen per pilot worden geïtereerd.

De eerder beschreven pilots neem ik per pilot op in de planning. Er is te zien dat ik eerst pilot 1 uitvoer. Hierna heb ik een workshop, een hackaton opgenomen in de planning. Dit is een type workshop die vaker wordt gedaan bij Hyves. Meer hierover volgt in de beschrijving van de pilotontwikkeling.  
Na de workshop werk ik de andere 2 pilots verder uit. Hierna is de iteratievolgorde opgenomen, zoals beschreven in de activiteit *definieer ontwikkelscenario*.

### Fase 3: Invoering

Activiteit 1: Stel plan van aanpak op   
Activiteit 2: Stel meetplan op  
Activiteit 3: Voer pilots in  
Activiteit 4: Draag pilots over

Het laatste onderdeel dat opgenomen is in de planning is de invoering. Enkele onderdelen in de planning, zoals een presentatie op Hyves, of de invoering voor de testgroep, heb ik na het ontwikkelen van het testscenario toegevoegd aan de planning.

# 2. Deel II

In deel twee van het afstudeerverslag beschrijf ik hoe ik per pilot de drie fases van de IAD-methode doorloop, respectievelijk *de definitiestudie*, *de pilotonwikkeling* en *de invoering*. In dit deel zal ik de onderzoeksvraag beantwoorden en beschrijven hoe ik dat gedaan heb. De structuur die ik hierbij aanhoud is *Wat heb ik gedaan?*, *Hoe heb ik dat gedaan?* en *Waarom heb ik dat zo gedaan?*.

## Fase 1: Definitiestudie

De eerste fase van het project is volgens de IAD methode de definitiestudie. Het doel van deze fase is het analyseren van de doelen van het systeem. Het is een initiatiefase waarbij ontwikkelaar, opdrachtgever en eventueel gebruiker samen gaan bepalen wat het project moet opleveren en hoe dat gedaan moet worden. De definitiestudie levert ook een systeemconcept op, wat al een globale weergave is van het prototype.  
De definitiestudie is opgedeeld in een lijst met activiteiten. Deze activiteiten hebben allemaal een apart doel en worden met bepaalde technieken behandeld. Elke activiteit levert een of meerdere producten op. De fase beschrijf ik, door per activiteit aan te geven wat het doel is, wat welke technieken moeten worden, hoe ik deze heb gebruikt en welke producten dat heeft opgeleverd.

Activiteit 1: Stel plan van aanpak op

Het doel van deze activiteit is tot afspraken te komen over de inhoud en de omvang van de op te leveren producten en de daarvoor uit te voeren activiteiten. In de definitiestudie wordt een globale planning gemaakt aan de hand van deze activiteiten. Er wordt vastgelegd wie er gelieerd zijn aan de afstudeeropdracht en wat hun rollen zijn. Om deze activiteiten te succesvol volbrengen schrijft de IAD methode enkele technieken voor, namelijk: Analyse van cruciale succesfactoren, Vastlegging organisatiestructuur en planningstechnieken.

Dit plan van aanpak is een detaillering van het algemene plan van aanpak en is toegespitst op de nu te doorlopen fase, de definitiestudie. De IAD methode beschrijft dat het plan van aanpak een product is dat bij elke activiteit iets uitgebreid is. Bij de eerste iteratie is er natuurlijk nog niet zoveel informatie over het project, en de activiteiten die moeten worden doorlopen. Veel van die informatie wordt duidelijk bij het doorlopen van de tweede activiteit van de definitiestudie, het definiëren van het ontwikkelscenario.

Ik maak het plan van aanpak door verschillende deelactiviteiten die de IAD methode voorschrijft voor het plan van aanpak te doorlopen, namelijk:

* + Bevestiging reikwijdte definitiestudie
  + Bepaal resultaten definitiestudie
  + Beschrijf projectstructuur
  + Maak tijdschatting en faseplanning

Activiteit 2: Definieer ontwikkelscenario

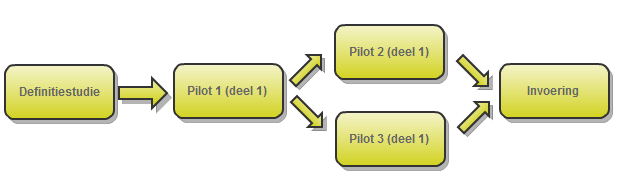
Deze activiteit heeft twee hoofddoelen. Aan de ene kant is een doel van deze activiteit de reikwijdte en de context van het project te bepalen. De activiteit geeft eigenlijk antwoord op de vraag wat er precies voor product moet komen. Aan de andere kant wordt de iteratiestrategie beschreven.  
De belangrijkste techniek voor het eerste deel van de activiteit is de workshop. In deze workshop wordt er geanalyseerd wat de cruciale succesfactoren zijn. Als alle gecommitteerden meedoen aan deze workshop, wordt ervoor gezorgd dat de verwachtingen van alle partijen worden bepaald. Dit voorkomt misverstanden of teleurstellingen  
Bij Hyves is het gebruik bij een nieuw product of project dat er een RFC, een *Request For Comments*, wordt aangemaakt. Dit RFC vindt plaats op het intranet en gecommitteerden worden gevraagd hierin hun ideeën en visies te geven.  
Het RFC voor Hyves Voorspellen levert een lijst met punten op die overeen is gekomen.

* Hyves Voorspellen moet het mogelijk maken om voetbalwedstrijden te kunnen voorspellen.
* Hyves Voorspellen zal worden gebouwd bovenop de huidige codebase
* Hyves Voorspellen zal eerst alleen over voetbal gaan, maar kan eventueel later worden uitgebreid met andere onderwerpen als de Tour de France of Idols.
* Er moeten verschillende competities kunnen worden voorspeld.
* Hyves Voorspellen moet een competitief systeem worden, waarbij gebruikers beter willen zijn dan hun vrienden.
* Viraliteit moet worden gezocht in het doorsturen van activiteit naar vrienden

Het tweede deel van deze activiteit is het beschrijven van de iteratiestrategie. In hoofdstuk 1.8 heb ik de strategie uitgebreid beschreven. De beschrijving in deze activiteit spitst zich toe op het praktische deel van de iteratiestrategie van dit project. Ik zal uiteenzetten welke pilots ik ga uitvoeren en wanneer ik dat ga doen. Ik kan dit nu doen omdat ik een beter beeld heb van het project, dan aan het begin.

De definitiestudie van de eerste iteratie is de belangrijkste, en neemt het meeste tijd in beslag. Ik heb deze definitiestudie in de planning opgenomen. Na de definitiestudie wil ik als eerste pilot de databases maken. Ik wil deze pilot als eerste doen omdat dit een pilot is die ik kan doen, zonder dat ik direct informatie nodig heb die uit andere pilots voortkomt.  
Hierna wil ik de eerste iteratie van de pilots ‘front end’ en ‘back end’ doen. Deze pilots doe ik naast elkaar omdat ze informatie van elkaar nodig hebben.  
  
Ik wil hier kort een voorbeeld van geven uit systeemeis 1a:  
“Er moet een overzichtspage/homepage zijn, waarvandaan naar alle wedstrijden kan worden genavigeerd.”

Ik wil dus een overzicht van de te voorspellen wedstrijden laten zien aan de gebruiker. Om de front end hiervoor te maken heb data nodig om te laten zien. Deze data moet uit de database komen en de back end moet die informatie uit de database halen en aan de front end geven. Ik ontwikkel hier dus de back end en de front end tegelijk.  
  
Deze situatie komt vele malen voor bij het ontwikkelen van een softwaresysteem.  
Als ik deze twee pilots heb ontwikkeld, wil ik de laatste fase starten, de invoering. Deze invoering is bedoeld om met de gebruikerstesten / controlegroep te kunnen beginnen. De eerste iteratie is nu compleet:

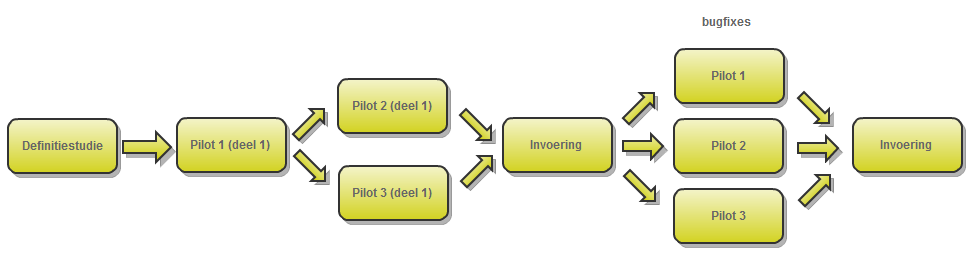


**Inzet: eerste iteratie van de ontwikkelcyclus.**

Na de invoering wil ik zoveel mogelijk feedback verzamelen van de controlegroep. Deze input verzorgt de belangrijkste informatie voor de volgende definitiestudie.

*‘Release Early, Release Often’ en bugs*.  
Ervaring leert dat hoe goed je ook test, er zullen toch veelal softwarefouten (bugs) voorkomen. Dit is zeker het geval als de RERO strategie wordt aangehouden. Nu zou iemand kunnen zeggen dat een strategie niet goed is als de kans groter is dat er bugs kunnen ontstaan. Dit is echter een afweging die moet worden gemaakt. De website van een bank hoeft bijvoorbeeld niet snel te evolueren, maar mag absoluut geen fouten hebben. Dat kan immers grote financiële gevolgen hebben. Bij een sociaal netwerk is dit niet het geval, maar is het wel erg belangrijk dat er zo snel mogelijk aan de wensen van gebruikers wordt voldaan, of gereageerd wordt op ontwikkeling van concurrenten. Er wordt daarom bij Hyves veel gedaan om bugs te voorkomen, maar er wordt niet zoveel getest als bij een bank. Het voordeel is een wekelijkse release met nieuwe functionaliteit tegenover de website van een bank die maar 1 a 2 keer per jaar een nieuwe softwareversie uitbrengen.

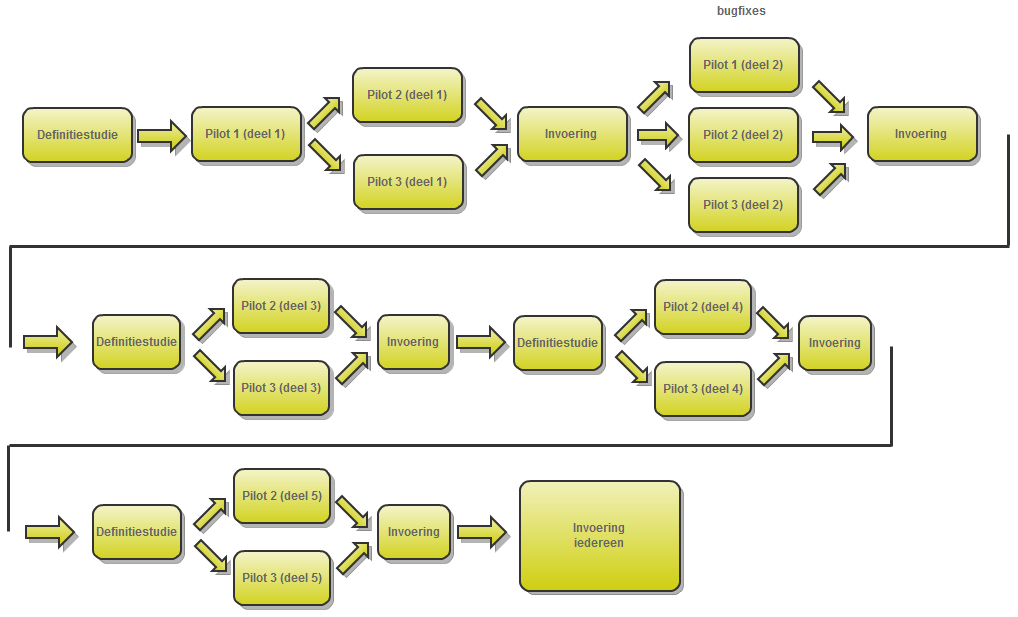
Om deze reden wil ik na de eerste invoering een iteratie doen, zonder definitiestudie, waarbij ik bugs wil oplossen, en deze oplossingen in wil plannen voor de release de week erop.



**Inzet: eerste en tweede iteratie van de ontwikkelcyclus.**

Hierna volgt de tweede definitiestudie, waarbij de input van de testgroep gebruikt wordt om aanpassingen aan het systeem te formuleren. Deze aanpassingen aan het systeem zal ik in een volgende pilotontwikkelingfase implementeren. In deze fase neem ik alleen pilot 2 en 3 op, omdat ik niet verwacht dat er nu nog aanpassingen gedaan moeten worden aan het datamodel. Mocht er nieuwe functionaliteit moeten worden gemaakt die toch een aanpassing van het datamodel vereisen, kan ik een nieuw deel van pilot 1 in een cyclus opnemen. De korte cyclus van een week, die ook weer afgesloten wordt met een invoering, wordt hierna nog twee keer herhaald. Deze iteraties zijn alle drie bedoeld om aanpassingen te implementeren die uit de gebruikerstesten zijn gekomen. Ze hebben dus nog niet een specifiek onderwerp. Na de laatste iteratie begin ik aan het PVA van de invoeringsfase. De week erna zal Hyves Voorspellen open gaan voor iedereen.

Het totale scenarioplan ziet er als volgt uit:



**Inzet: Totale scenarioplan.**

Activiteit 3: Definieer systeemeisen

Het doel van deze activiteit is het opleveren van een lijst van systeemeisen. Aan de hand van deze systeemeisen zal het product worden ontworpen. De systeemeisen geven concreet referentiemateriaal bij het uitvoeren van de volgende activiteiten.  
De voorgeschreven techniek voor deze activiteit is de workshop.  
Deze workshop houd ik in de vorm van een gesprek met de opdrachtgever. We kijken naar de input van de RFC uit het ontwikkelscenario en bespreken hoe we dit praktisch in kunnen richten in de website. Dit gesprek levert een groot aantal punten en wensen op. Ik prioritiseer deze met behulp van de MoSCoW methode. Hierbij komen alleen de Must-have punten in de systeemeisen. Deze zijn immers echt vereist voor het systeem. De eerste versie van de systeemeisen is nog niet erg gedetailleerd. Dat is geen probleem, hoe verder het project is, hoe gedetailleerder de systeemeisen worden. De IAD methode beschrijft dat er bij het opstellen van de systeemeisen rekening moet worden gehouden met ‘Scope Creep’. Dat betekent dat er requirements bij komen gedurende het de uitvoering van het project. Bij sommige strategieën is dit absoluut niet de bedoeling. Bij de strategie die ik aanhoud is het juist de bedoeling dat er nieuwe requirements bij komen.

**Inzet: Eerste versie systeemeisen**

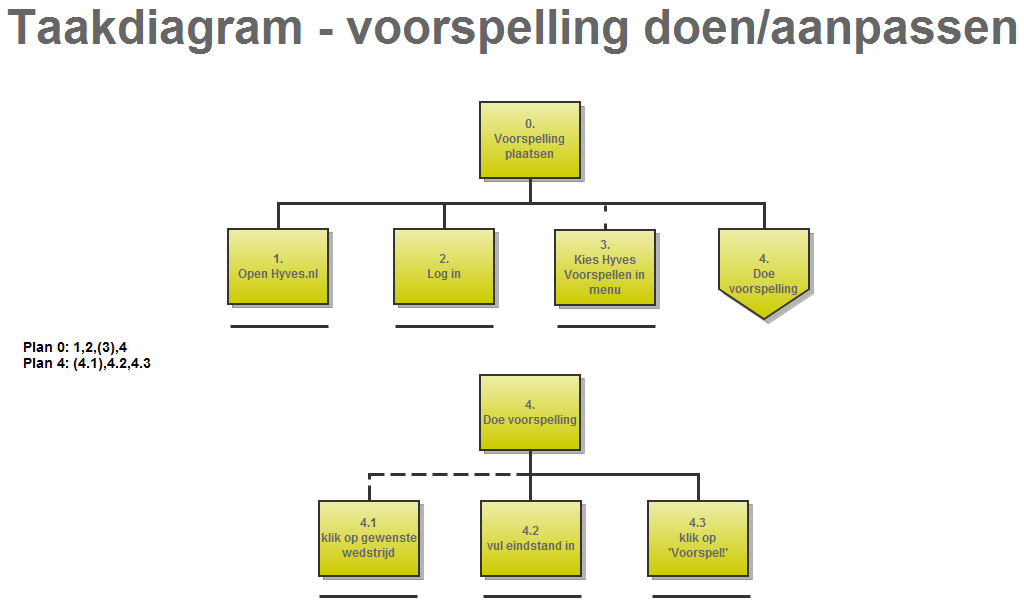
1. Hyves voorspellen manifesteert zich op 3 plekken binnen Hyves:
   1. Een overzichtspagina, waar vandaan naar alle wedstrijden kan worden genavigeerd. (Hyves Voorspellen homepage)
   2. Een itempagina, waar alle acties rondom het event te doen zijn.
   3. De homepage, waar gebruikers die meedoen proactief worden uitgenodigd en uitgedaagd om te voorspellen. Ook updates van voorspellingen zijn te zien op de homepage (`buzz-items`).
2. Op de homepage moet een `blok` zichtbaar worden, enkele dagen voor een bepaalde wedstrijd, waar die wedstrijd direct is te voorspellen. Ook moet te zien zijn of vrienden deze wedstrijd al voorspeld hebben.  
   Na de wedstrijd moet dit blok tonen wat de uitslag is en welke vrienden dat goed hadden.
3. Als een voorspelling gedaan wordt, moet dit als update gedistribueerd worden naar de homepage van vrienden van diegene (buzz-item).
4. De Hyves Voorspellen homepage moet buiten de te voorspellen wedstrijden weergeven, ook laten zien wat de status is van het systeem. Dit houdt in dat te zien is hoeveel punten of andere beloningen diegene heeft verzameld.
5. De itempagina moet een pagina worden die volledig op dat event of wedstrijd is toegespitst:
   1. De voorspelling moet daar te doen zijn, en ook kunnen worden aangepast.
   2. Er moet te zien zijn wat anderen voorspellen voor deze wedstrijd.
   3. Er moet kunnen worden gepraat over de wedstrijd.
   4. Na de wedstrijd moet de uitslag van de wedstrijd en beloningen te zien zijn

Activiteit 4: Bepaal systeemconcept

Deze activiteit is een vervolg van het definiëren van de systeemeisen en heeft als doel de functionele architectuur van het systeem voor ogen te krijgen. Het systeemconcept is het product gezien vanuit het perspectief van de gebruiker, ofwel *actor*. Het is belangrijk om de gebeurtenissen of *events* die de actor kan doen op een rijtje te krijgen, zodat je kunt beslissen hoe het systeem daarmee om dient te gaan.   
 De te gebruiken techniek heeft voor ogen om de beschrijving van events te kunnen faciliteren. Taakdiagrammen zijn uitermate geschikt om userflows te beschrijven. Nadat deze *flow* duidelijk is, kan deze worden gevisualiseerd met behulp van wireframes. Om een en ander te verduidelijken beschrijf ik in een apart document wat bepaalde elementen in de wireframes precies zijn.  
  
Door de systeemeisen te definiëren is al grotendeels bekend geworden welke pagina’s zullen worden toegevoegd aan Hyves, met welke inhoud. Ik maak taakdiagrammen voor acties die de gebruiker kan uitvoeren uit de systeemeisen. Deze acties zijn:

* Een voorspelling doen of aanpassen.
* Een reactie plaatsen bij een wedstrijd, of bij een voorspelling.
* Een vriend uitdagen om ook een wedstrijd te voorspellen.
* Het clubprogramma raadplegen.

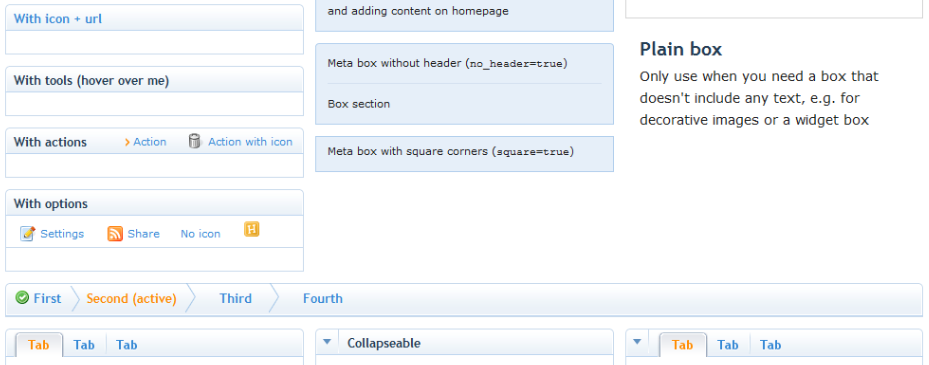
**Inzet: taakdiagram 1**

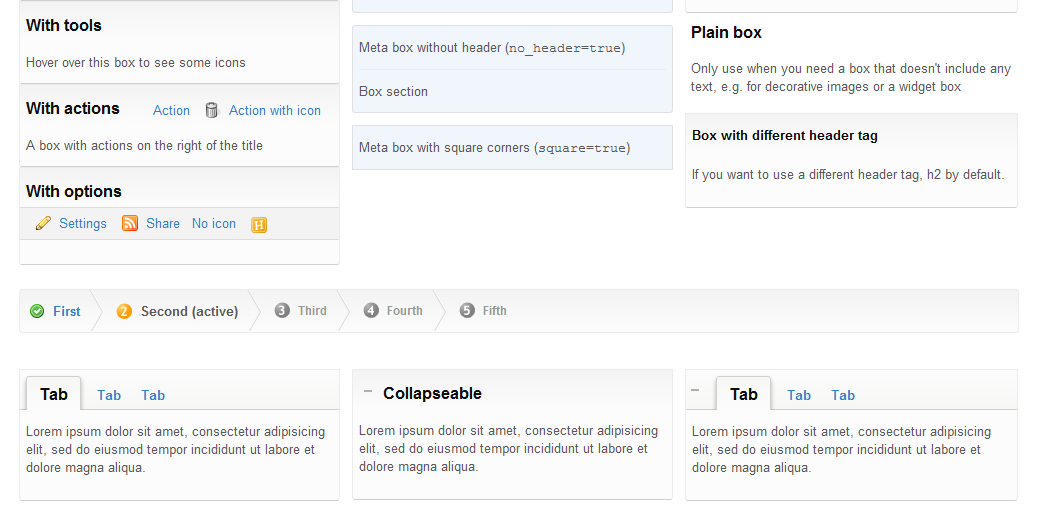


Ik wil iets vertellen over hoe de taakdiagrammen, ‘voorspelling doen/aanpassen’ in het specifiek tot stand zijn gekomen. Met een taakdiagram wordt de flow duidelijk die gebruikers moeten volgen om een bepaalde actie te voltooien. Ik kijk bij het maken van een taakdiagram vanuit het perspectief van een gebruiker. Ik houd hierbij aan de ene kant in gedachte wat ik samen met de opdrachtgever besproken heb voor de systeemeisen. Aan de andere kant probeer ik een flow te ontwikkelen waarbij ik de 5 e’s respecteer (effectief, efficient, engaging, error tolerant en easy to learn).  
Het startscherm van Hyves is de ‘pulse’ of ‘buzz’ pagina. Dit is de pagina die een gebruiker te zien krijgt als hij naar Hyves.nl gaat en inlogt. Om mee te kunnen doen met Hyves Voorspellen moet de gebruiker ingelogd zijn. Dit verklaart stap 0, 1 en 2 van het diagram. Deze stappen zijn bij elk taakdiagram hetzelfde.  
Stap 3 is het navigeren naar de Hyves Voorspellen homepage en is niet per se vereist om een voorspelling te doen. We hebben immers in de systeemeisen besloten dat er op de Hyves homepage een blok komt waarmee ook een voorspelling gedaan kan worden. Dit verhoogt de effectiviteit en brengt het product extra onder de aandacht.  
Stap 4 is het doen van de voorspelling. Als de gebruiker stap 3 heeft gedaan en op de Hyves Voorspellen homepage is, moet de gebruiker nu de wedstrijd aanklikken die hij wil voorspellen, om op de wedstrijdpagina te komen. Dat is stap 4.1 en is niet nodig als de gebruiker nog op de Hyves Homepage is, omdat de gebruiker dan al het voorspellingsformulier voor zich heeft.  
Op dit moment heeft de gebruiker een formulier voor zich, waar hij de score in kan vullen, stap 4.2. Het formulier wordt verstuurd als de gebruiker op ‘voorspel!’ klikt, stap 4.3.

Het wijzigen van een voorspelling gaat precies op dezelfde manier. Het enige verschil is de status van het systeem. Deze flow vergt weinig muisklikken en is daarom effectief. Door het voorspelformulier op de Hyves Homepage aan de gebruiker te laten zien is het ook gemakkelijk in gebruik. Error tolerantie, engaging en efficientie zal ik later ik het verslag behandelen.

*Design patterns en mockups*Nadat ik de taakdiagrammen heb gemaakt en redelijk duidelijk is geworden wat gebruikers kunnen doen op de verschillende pagina’s, kan ik deze pagina’s een eerste vorm gaan geven.   
Ben Shneiderman, een van de grondleggers van Interaction Design, heeft 8 ID regels opgesteld. De eerste en wellicht belangrijkste regel is consistentie. Als een systeem ontwikkeld wordt, dient het consistent te zijn in zowel de werking, de bediening als ook het uiterlijk.  
Om deze reden is het design van Hyves is vastgelegd in bepaalde patterns. Deze design patterns zijn erg belangrijk om de consistentie van design op Hyves te kunnen garanderen. Ook kan er door deze patterns te stylen, direct een grafische aanpassing door heel Hyves worden gedaan. Ik heb een voorbeeld hiervan in dit verslag opgenomen:

  
**Inzet: Design patterns op Hyves (voor redesign september ‘11).**

**Inzet: Design patterns op Hyves (na redesign september ‘11).**

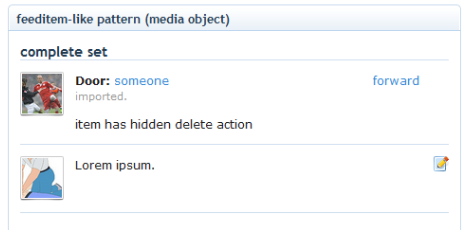
Door het gebruik van deze patterns is het grafische design in dit project niet zo belangrijk, dit staat immers vast. Het is wel belangrijk dat ik deze patterns gebruik om mijn pagina impressies (mockups) te maken. Om dit zo goed mogelijk te doen is er bij Hyves een manier ontwikkeld om snel een pagina neer te zetten die gebruik maakt van de vooraf gedefinieerde patterns. Het is zelfs zo handig dat je wireframes kunt maken die er bijna zo uitzien als ze uiteindelijk worden.  
  
Ik maak voor alle pagina’s die moeten worden gemaakt een mockup. Naar aanleiding van de systeemeisen moeten er drie hoofdposities/schermen worden ontwikkeld:

* Overzichtspagina (Hyves Voorspellen homepage)
* Blok op de Hyves homepage met aankomende wedstrijden
* Itempagina, de pagina van 1 wedstrijd



**Inzet: mockup van de Hyves Voorspellen homepage.**

Ik wil beschrijven hoe het overzichtsscherm tot stand is gekomen. Uit de systeemeisen en de taakdiagrammen is gebleken welke acties op welke pagina’s moeten gebeuren. Het belangrijkste op de pagina zijn de links naar events. Gebruikers moeten vanuit deze pagina navigeren naar de itempagina’s. Omdat dit de belangrijkste functie is van deze pagina, zet ik deze links linksboven, wat de eerste plek is die gebruikers lezen als ze op een website komen. De vorm die dit link lijstje heeft is een pattern op Hyves, en heet media-object list.

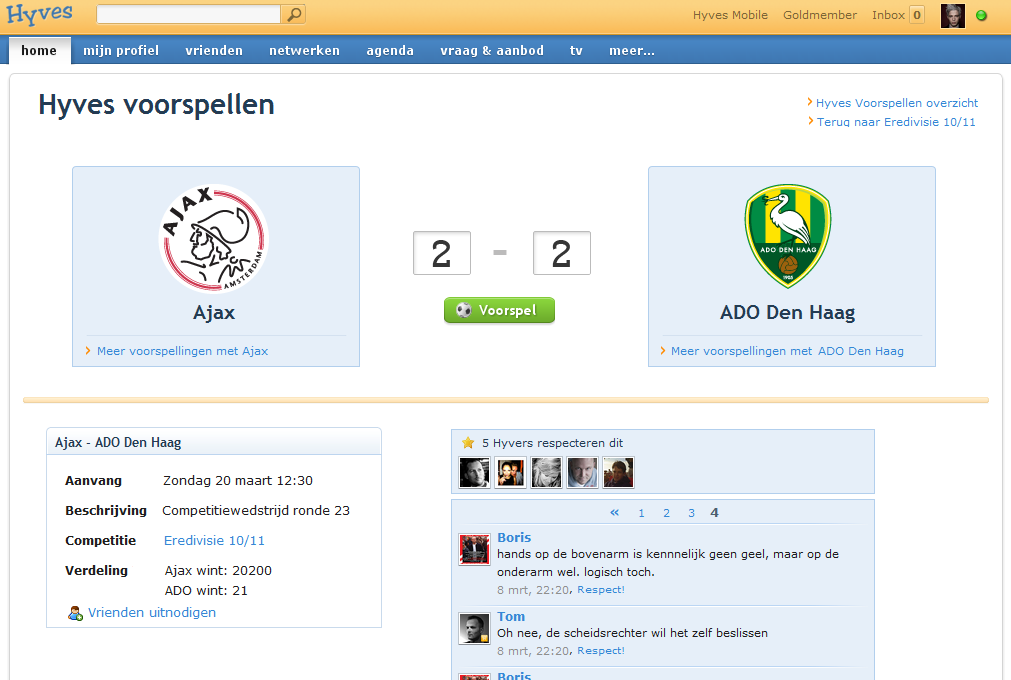


**Inzet: media object pattern uit pattern documentatie.**

In deze lijst heb ik overigens ook al rekening gehouden met het gebruik van meerdere competities. Dit staat niet in de systeemeisen en is dus niet vereist, maar omdat Hyves voorspellen later wellicht wordt uitgebreid, wilde ik het toch meenemen in het ontwerp.

In de systeemeisen staat ook dat de status van het systeem op deze pagina te zien moet zijn. Deze systeemstatus is het aantal beloningen die de gebruiker heeft verzameld. Er is nog niet bepaald wat die beloningen precies inhouden. In een wireframe maakt dat ook niet uit, het is geen systeemeis. Om het blokje beloningen toch een visuele invulling te geven heb ik gekozen voor badges. Badges zijn effectief in het duidelijk maken van de ‘5 primary functions of achievements’ (Goal Setting, Instruction, Reputation, Status/Affirmation en Group identification). Het is veel werk om het gebruik van badges goed te doen, maar voor de mockup is het prima.

Omdat de systeemstatus een veel kleiner oppervlak beslaat dan de links naar de itempagina’s is er ruimte voor andere functionaliteit. Ik heb hiervoor punten van de MoSCoW lijst gebruikt die ik met de opdrachtgever heb opgesteld, maar (nog) niet in de systeemeisen terecht zijn gekomen. Ik heb hiervoor een extra blok voor gedetailleerde systeemstatus, ‘jouw laatste weddenschappen’ gebruikt en een navigatieblok ‘andere competities’. Deze staan dus niet in de systeemeisen en worden wellicht niet, of niet in de eerste iteratie gemaakt.  
Ik volg dezelfde manier van ontwerpen naar prioriteit en het gebruik van de patterns om mockups te maken voor de andere onderdelen.

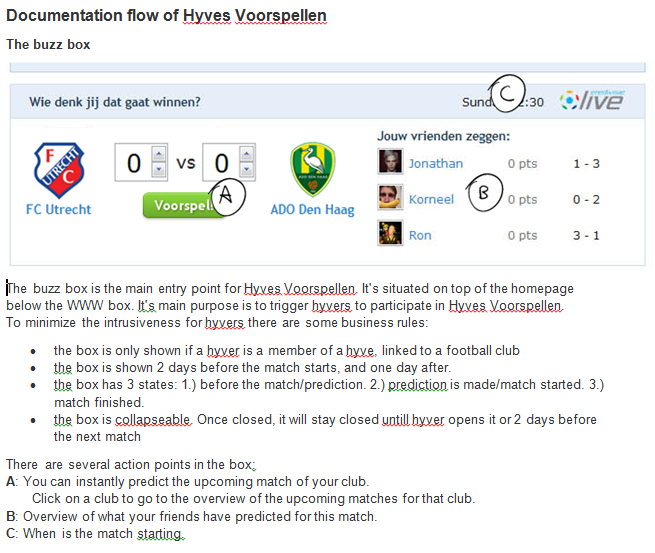


**Inzet: mockup van een item pagina.**

**Inzet: mockup blok op Hyves Homepage.**  
Naast de mockups zelf, maak ik een beschrijving die in detail laat zien wat de onderdelen op de mockups doen. De documenten samen moeten een duidelijk beeld geven van het concept dat ontwikkeld moet worden en is een documentatie van hoe het systeem dient te werken. Uiteraard kunnen onderdelen van de beschrijving in een latere iteratie herzien of aangepast worden. De beschrijving is toegevoegd als bijlage E. Het document is in het Engels aangezien dat de voertaal is bij Hyves. Het bevat uitleg en gaat dieper in op de systeemeisen.



**Inzet: onderdeel uit de beschrijving van een mockup.**



Activiteit 5: Beschouw technische structuur

Het doel van deze activiteit is het toetsen van de technische eigenschappen van het systeemconcept aan de bestaande technische structuur. Het heeft geen zin om een systeemconcept neer te zetten dat met geen mogelijkheid kan worden gerealiseerd in de technische omgeving.   
  
Bij een jonge ontwikkelomgeving is het belangrijk om te kijken of alle nieuwe functionaliteit wel is te realiseren in de huidige structuur. Bij Hyves ligt dit anders. Er is een volwassen, ontwikkelvriendelijke omgeving, met veel capaciteit. Dit maakt de omgeving niet per definitie makkelijker om mee om te gaan, maar wel een stuk krachtiger. Deze activiteit gaat er dan ook vooral om met welke aspecten rekening moet worden gehouden bij het ontwikkelen van het prototype van Hyves Voorspellen.   
De technieken die hiervoor gebruikt moeten worden zijn *architectuurontwerp* en *analyse van de technische structuur*.   
Voor deze beschouwing kijk ik naar verschillende aspecten, namelijk programmeertaal, ontwikkelcyclus en database architecturen. Ik kies het aspect *programmeertaal* omdat de programmeertaal een fundamenteel onderdeel is van het ontwikkelen en ik code in die taal moet kunnen schrijven. Ik kies ontwikkelcyclus, omdat dit een belangrijk onderdeel is van de invoering van het systeem. Keuzes voor de planning van mijn scenarioplan hebben een nauwe relatie met de ontwikkelcyclus op Hyves. Als laatst kies ik de database architectuur, omdat ik moet zorgen dat de database architectuur de nieuwe *load* aan kan.

*Programmeertaal*Allereerst kijk ik naar de programmeertaal. Hyves maakt gebruik van verschillende programmeertalen. Voor de front-end wordt uiteraard gebruik gemaakt van HTML, CSS en JavaScript. Om de abstractielagen apart te houden wordt er gebruik gemaakt van Smarty als koppeling tussen front- en backend. Voor de back-end wordt voornamelijk gebruik gemaakt van PHP, maar ook Python en C++ komen aan bod. Python wordt vooral voor automatische testen gebruikt op Hyves en C++ wordt gebruikt voor een compiling systeem (HipHop). Deze twee talen zijn voor mij dus niet aan de orde. Hyves maakt gebruik van HipHop, dat PHP compiled naar machinetaal, dat het een stuk sneller maakt. Om deze reden wordt alle back end code in PHP geprogrammeerd en is dit ook de programmeertaal die ik ga gebruiken.

*Ontwikkelcyclus*  
bij Hyves wordt wekelijks nieuwe functionaliteit *live (in productie)* gezet. Om te dat proces te begrijpen is het goed om te weten hoe de infrastructuur op Hyves er uitziet.  
Allereerst is er de development omgeving; Dat is een centrale database met dummy data. Iedereen heeft op zijn pc een versie van de Hyves codebase en een virtuele server, waar op ontwikkeld wordt.   
Als er een onderdeel is geprogrammeerd of een design is aangepast, worden deze aanpassingen naar een *staging omgeving* gezet. Dit is een omgeving met een centrale database en een centrale webserver. Hier kunnen alle aanpassingen worden getest voordat ze live worden gezet.  
Als laatst is er de live - of productie - omgeving, die o.a. bestaat uit honderden webservers en centrale databases verspreid over verschillende databaseserverclusters. In totaal heeft Hyves meer dan 2000 servers.  
  
Nieuwe functionaliteit wordt ontwikkeld en getest op dev, getest op staging en vervolgens naar de live servers gezet. Dit laatste gebeurt volgens een vaste cyclus. Elke kwartaal wordt alle development code live gezet. Alle code van de ontwikkelmachines wordt dan getest en naar live gedeployed, dit heet een *major release (of trunk release)*. Wekelijks wordt de staging code getest en live gezet, een *minor release*. Dagelijks zijn er releases met bugfixes, *stable releases.* De trunk releases zijn vooral bedoeld om aanpassingen te doen aan grote databases. Hyves Voorspellen kan mee met een minor release, omdat het allemaal nieuwe code en databases zijn, die de andere databases nauwelijks beïnvloeden. De functionaliteit uit de tweede, derde en vierde iteraties van de pilots zullen worden ingevoerd met een minor release.

*Database architecturen*Naast de programmeertaal en de ontwikkelcyclus is het belangrijk om te weten hoe er met databases wordt gewerkt. Dit kan namelijk snel het tijdrovende deel van de site worden en snelheid is een van de belangrijkste aspecten van webdevelopment.

De databases op Hyves zijn verdeeld over enkele honderden servers, die opgedeeld zijn in clusters, of architecturen. Zo is er bijvoorbeeld een groepje servers dat alle gegevens m.b.t. financiële gegevens bevat en een groepje servers dat alle gadget gegevens bevat. Ik wil weten welke architectuur ik kan gebruiken. Ik heb hiervoor een senior software engineer en een senior system engineer uitgenodigd voor een gesprek. In dit gesprek heb ik uitleg gegeven over wat de bedoeling is van Hyves Voorspellen en wat de functionaliteiten ongeveer zullen zijn. In dit gesprek hebben we besloten dat we voor Hyves Voorspellen de architectuur zullen gebruiken die nog de meeste ruimte over heeft voor extra load. Voor Hyves Voorspellen zullen we de *profile architectuur* gaan gebruiken.

Activiteit 6: Beschouw organisatorische inrichting

Het doel van dit onderdeel is het bekijken wat de impact is op de organisatie. Of er veranderingen zijn in workflow of bepaalde acties van mensen. Aangezien Hyves Voorspellen een autonoom, zelfvoorzienend programma dient te worden, is de impact op de bedrijfsprocessen niet zo groot.

Aan het product Hyves Voorspellen wordt een Product Owner(PO) toegewezen. Deze PO is verantwoordelijk voor de continue werking van het product. Zo moet het systeem aan het begin van een competitie worden geconfigureerd. De data voor wedstrijden en uitslagen komen binnen via een soort RRS feed. Dit is dus een geautomatiseerd proces.   
Om de blijvende werking van het systeem te kunnen garanderen, zullen geautomatiseerde regressietesten moeten worden gemaakt. Voor elk product op Hyves zijn zulke testen. Het is de verantwoordelijkheid van degene die de release leidt om deze testen in de gaten te houden.

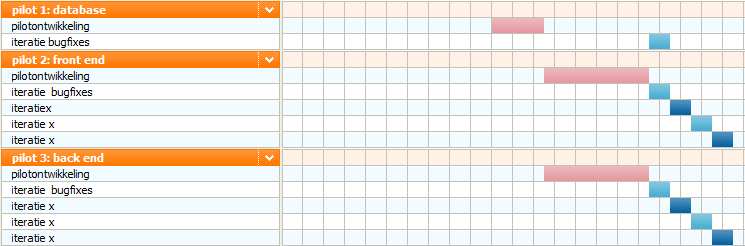
Activiteit 7: Stel pilotplan op

Het doel van deze activiteit is een geactualiseerde versie van het pilotplan op te leveren. In het hoofdstuk ‘definieer ontwikkelscenario’ heb ik duidelijk beschreven hoe ik de pilots uit ga voeren.

Het pilotontwikkelplan met iteratiestrategie ‘hybride ontwikkelen’ is opgebouwd uit 3 pilots:

* Pilot 1: Database
* Pilot 2: front end
* Pilot 3: back end

Ik zal eerst 1 iteratie van elke pilot doorlopen. Hierna komt de gebruikerstestgroep in beeld. Vervolgens worden de volgende iteraties van de pilots uitgevoerd zoals in de planning staat



**Inzet: Pilots in de planning.**

Ik zal allereerst het datamodel ontwikkelen. Dat doe ik vanuit de systeemeisen en het systeemconcept. Ik heb hier geen andere input voor nodig. Ik heb dit datamodel wel nodig om aan de slag te kunnen met de andere pilots.

De volgende pilots die ik ga uitvoeren zijn de front end en de back end. Ik heb in het definiëren van het ontwikkelscenario duidelijk omschreven waarom ik deze nu parallel uitvoer.  
Hierna itereer ik de pilots 2 en 3 nog 3 keer met de input van gebruikers.

De eerste iteratie van de drie pilots hebben in mijn opzet geen deel of een subset van de systeemeisen als onderwerp. Pilot 1 is het ontwikkelen van het datamodel. Pilot 2 is het realiseren van de front end en deel 3 is het realiseren van de back end. Uiteraard is er per pilot een bepaalde aanpak nodig om het goed uit te voeren. Deze behandel ik in de fase ‘pilotontwikkeling’ per pilot in het plan van aanpak.

Fase 2: Pilotontwikkeling

Het doel van deze fase is het ontwikkelen van de pilots. In de definitiestudie is beschreven welke pilots er zijn en is er een pilotplan gemaakt. Aan de hand van dit pilotplan, maar vooral ook het ontwikkelscenario wordt deze fase doorlopen. Het is belangrijk om te onthouden dat deze pilots iteratief zijn en veelal niet sequentieel maar parallel worden uitgevoerd. Er is na het definiëren van het systeemconcept al een vrij goed beeld ontstaan van hoe het prototype er uit moet komen te zien.

Het IAD ontwikkelmodel beschrijft vier hoofdbestanddelen van de fase pilotontwikkeling, die ik als activiteit zal uitvoeren:

* + Plan van aanpak
  + Ontwerp software bouweenheden
  + Bouw software bouweenheden
  + Beoordeel en test pilotdeel

De drie pilots die ik heb gedefinieerd moeten deze 4 activiteiten dus bevatten. Ik beschrijf per pilot deze activiteiten en zal per pilot een hoofdstuk aanmaken.

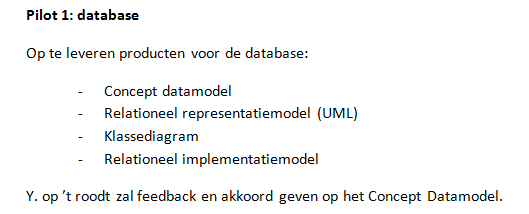
### Pilot 1: Database, deel 1

De eerste pilot die ik doorloop is het ontwerpen en ontwikkelen van de database.

#### Activiteit 1: Plan van aanpak

Het doel van deze activiteit is duidelijke afspraken vast te leggen over de op te leveren producten en de daarvoor uit te voeren werkzaamheden.   
Het maken van het datamodel is een activiteit die ik voornamelijk zelf zal uitvoeren. Ik zal feedback krijgen van een senior software engineer bij Hyves, om de kwaliteit te waarborgen.

In dit geval is het plan van aanpak vooral een leidraad om de benodigde producten op te leveren.



**Inzet: vereiste producten voor pilot 1 in PVA.**

#### Activiteit 1: Ontwerp software bouweenheden

Het doel van deze activiteit is het specificeren van de technische structuur van de software-bouweenheden voor deze pilot. Deze activiteit is onderdeel van de pilot ‘Database’. Daarom gaat het hier om het ontwerpen van de database. De technieken bij deze activiteit zijn Entiteit/relatie-modellering en procesbeschrijving.

##### Concept datamodel ontwerpen

Een concept datamodel is een schematische weergave van de database op papier. Aan de hand van dit concept is te zien hoe de dataobjecten zich tot elkaar verhouden en of er zich geen technische problemen voordoen. Technieken die hiervoor gebruikt worden zijn *gegevensanalyse, Entiteit/relatie-modellering, procesbeschrijving* en *workshops*.  
Naar aanleiding van het systeemconcept stel ik een concept datamodel op. Hierbij heb ik de functionaliteit die aanwezig dient te zijn in het prototype verzameld en dit genormaliseerd naar benodigde objecten( of entiteiten) en de relatie onderling.

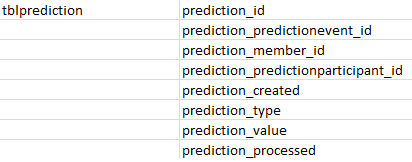
Om te laten zien hoe dit proces verloopt, kan ik het best een voorbeeld geven van een object en hoe het tot stand komt vanuit de systeemeisen. We nemen systeemeis 5a, de gebruiker is op de pagina van een wedstrijd en wil een voorspelling doen:

Je kunt hier de *wie, wat, waar* vragen bij stellen:

- Wie? Degene die de voorspelling doet.  
- Wat? De voorspelling.  
- Waar? De wedstrijd.

Er bestaat al een gebruikersobject op Hyves, dus die kunnen we gebruiken. Er bestaat nog geen voorspellingsobject, dus dat is het eerste nieuwe object. De wedstrijd is het tweede. De wedstrijd is nog verder te normaliseren: een wedstrijd heeft twee deelnemers. Dit is ons derde objectsoort.  
  
 Ik wil wat dieper ingaan op de opzet van het voorspellingsobject. De voorspelling is een nieuw object, dat een 1 op 1 relatie heeft met een gebruikersobject (al bestaand op Hyves) en een wedstrijd. Een gebruiker mag namelijk maar één voorspelling doen per wedstrijd. Voor deze koppeling zijn dus 2 ‘vreemde sleutels’ nodig. De ‘primaire sleutel’ van het voorspellingsobject is het object-id, wat het mogelijk maakt goede indices op de tabellen te zetten en ze te koppelen aan andere tabellen. Het is belangrijk om te weten wanneer de voorspelling is gedaan, dus heeft het ook een DATETIME attribuut met het creatiemoment. We hebben inmiddels vier attributen in het datamodel voor het voorspellingsobject.  
In een later stadium wil Hyves niet alleen voetbalwedstrijden kunnen voorspellen (aantal doelpunten), maar ook andere types voorspellingen doen, zoals wie er wint bij een talentenshow. Dit soort voorspellingen hebben een andere aard en hebben niet een 1 op 2 relatie (1 wedstrijd, 2 voetbalclubs), maar een 1 op n relatie(1 wedstrijd, n kandidaten). Om dit probleem te tackelen kies ik ervoor om een attribuut ‘type’ toe te voegen en een attribuut ‘value’, dat bijvoorbeeld het aantal doelpunten bevat. Bij deze opzet kan een voorspelling per persoon meerdere voorspellingsrijen in de database hebben.  
Als laatst voeg ik een kolom ‘processed’ toe. Deze kolom is voor het toekennen van punten voor weddenschappen en bevat een boolean (vlaggetje) als de weddenschap is verrekend in het puntenaantal van de voorspeller.

Het voorspellingsobject en de andere objecten zijn opgenomen in bijlage E: Het concept datamodel.



**Inzet: De concepttabel voor het voorspellingsobject.**

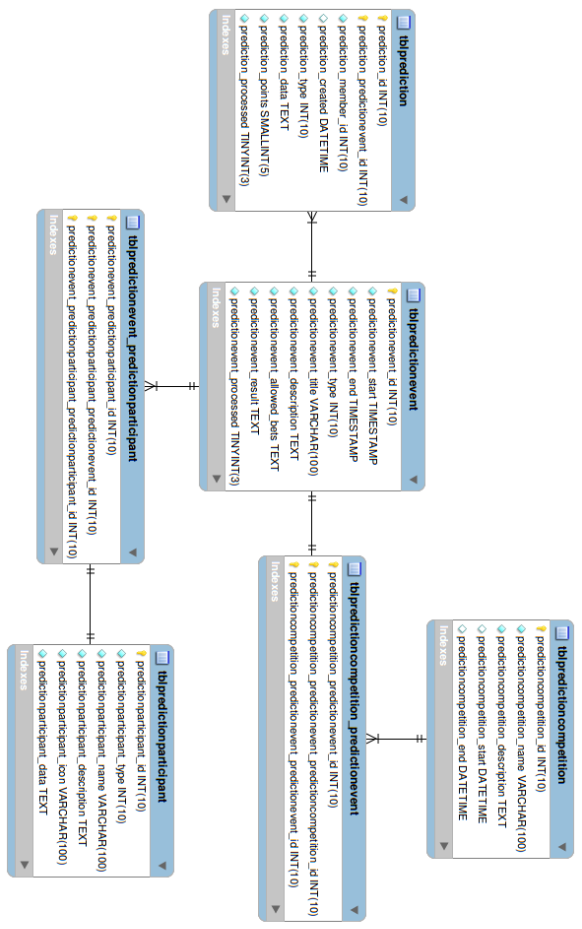
Workshop  
Aan de hand van het concept datamodel organiseer ik een kleine workshop met een senior software engineer bij Hyves, Y. op ‘t roodt. We bespreken het datamodel en kijken naar punten die beter kunnen.  
Een van de punten is de structuur van het type weddenschap en de ‘value’. Mijn idee om te normaliseren zorgt er zoals gezegd voor dat elke voetbalweddenschap 2 entries in de database heeft. Een andere opzet is het type en value in een ‘data’ kolom te zetten in een zogenaamde ‘property-bag’. Het voordeel hiervan is dat de database minder groot wordt, waardoor het sneller is. Het nadeel is dat de interpreter(php) de waardes moet gaan evalueren (interpreteren), wat ook tijd kost. Dit kan snel veel langer duren dan in een grotere database zoeken.  
Door een recente update in de Hyves codebase is het echter mogelijk geworden om de property-bag op een andere manier (native), veel sneller te evalueren. Hierdoor wordt het snelheidsprobleem van de property-bag opgelost en is dit de beste implementatie.

Ook voegen we een competitietabel toe, waardoor verschillende competities opgezet kunnen worden, zoals de Eredivisie en de KNVB Beker. Door middel van een koppeltabel worden de wedstrijden aan de competities gekoppeld.

Het concept is nu aangepast en is geen concept meer, maar een product. Dit product is het relationeel representatiemodel.

Klassediagram ontwikkelen

Het doel van een klassediagram is het visueel weergeven van de eigenschappen van- en de relatie tussen objecten. Zo kunnen ook eventuele beperkingen in het systeem opgespoord worden.  
Aangezien de workshop het volledige model heeft opgeleverd, bestaat deze sub activiteit alleen uit het invullen van het model in een UML programma en het toekennen van de juiste datatypes aan de attributen van de objecten. Het resultaat hiervan is een klassediagram dat op de volgende pagina is weergegeven.

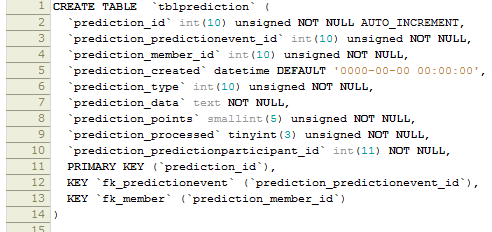


**Inzet: Klassediagram voor Hyves Voorspellen.**

#### Activiteit 2: Bouw software bouweenheden

Relationeel implementatiemodel

Het doel van deze sub activiteit is vrij logisch, het produceren van de database.  
Sommige UML-programma’s kunnen vanuit het ingevoerde model direct een relationeel implementatiemodel creëren, wat natuurlijk bijzonder handig kan zijn. Ik heb dit ook gedaan en dat heeft het model opgeleverd:



**Inzet: MySQL query om tabel te creëren. Getoond is de tabel voor voorspellingen.**

Dit relationeel implementatiemodel heeft de vorm van een MySQL script. Dit script creëert de daadwerkelijke databases als je het op de MySQL server uitvoert. Deze scripts voer ik dus uit op de verschillende ontwikkelomgevingen van Hyves.

#### Activiteit 3: Test en evalueer pilotdeel

Het evalueren van het pilotdeel is van toepassing op het concept datamodel. Dit heb ik zoals beschreven geëvalueerd met een senior software engineer. Een datamodel zelf hoef je niet te testen. Als het model in het UML programma staat en dat programma produceert de representatie zoals het in het concept model staat, dan klopt het.

### Pilot 2: Front end, deel 1

De tweede pilot die ik doorloop is het ontwikkelen van de front end voor Hyves Voorspellen.

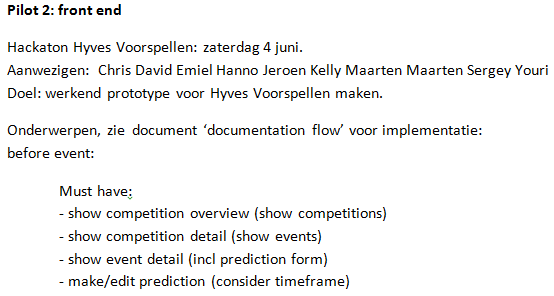
#### Activiteit 1: Plan van aanpak

Het doel van deze pilot is het vertalen van de mockups uit het systeemconcept naar front end code en grafische elementen. IAD schrijft enkele technieken voor, namelijk *gestructureerd programmeren* en *workshops*. Omdat deze pilot en pilot 3, de back end, tegelijkertijd worden ontwikkeld, worden deze ook tegelijkertijd getest. De activiteit ‘test en evalueer pilot’, zal ik daarom in het hoofdstuk *pilot 3, deel 1* behandelen.

*Hackaton*  
Ik wil deze techniekensamenvoegen in een hackaton. Een hackaton is een workshop waarbij tegelijkertijd en gezamenlijk aan een product gewerkt. Door deze samenwerking kan in korte tijd veel werk worden verzet. Omdat er veel experts tegelijk naar een product kijken, worden problemen snel opgelost en komen veel nieuwe inzichten of ideeën bovendrijven.

Na deze hackaton zal ik een inventarisatie maken van wat er gedaan is en bekijken wat er nog moet worden gedaan. Het plan van aanpak zal ik hierna aanpassen. Voor nu neem ik in het plan van aanpak informatie voor de hackaton op. Een hackaton moet niet al te strak geregisseerd zijn, maar ik heb verschillende onderwerpen uitgekozen waaraan deelnemers aan kunnen werken. Zij kunnen dan in het document ‘documentation flow’ zien hoe het te maken onderdeel er uit moet komen te zien en hoe het moet werken.

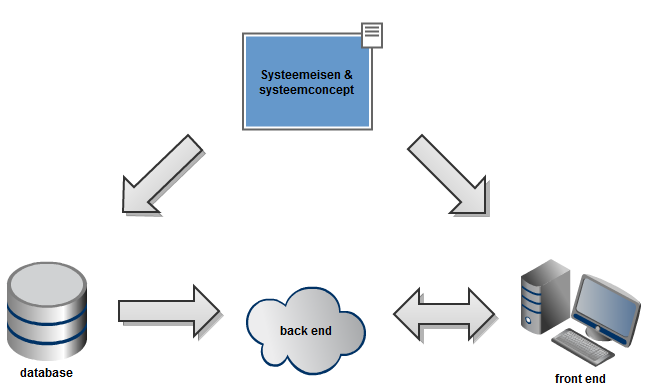
**Inzet: deel van het plan van aanpak voor pilot 2**



#### Activiteit 1: Ontwerp software bouweenheden

Het doel van deze activiteit is het specificeren van de technische structuur van de software-bouweenheden voor deze pilot. Er is bij het definiëren van het systeemconcept gekeken vanuit het perspectief van de gebruiker van het systeem. In deze activiteit gaat het om hoe de softwareontwikkelaars dit concept kunnen gaan realiseren. Dit alles met betrekking tot de pilot ‘Front end’. Te gebruiken techniek bij deze activiteit is procesbeschrijving.

Deze pilot gaat om het omzetten van het ontwerp naar een werkende front end. Dit is een proces dat tegelijkertijd wordt uitgevoerd met pilot 3, de back end. Deel 1 van pilot 2 en deel 1 van pilot 3 worden naast elkaar uitgevoerd, zoals ook beschreven in het ontwikkelscenario. Ik wil uitleggen waarom dit precies zo is, aan de hand van een figuur.



Inzet: volgorde ontwikkeling van pilots.

De systeemeisen en het systeemconcept zijn aan het begin van deze pilots beschikbaar. In hoofdstuk *pilot 1, deel 1* heb ik beschreven hoe vanuit het systeemconcept de database is ontwikkeld, voor dit proces staat de pijl linksboven. De database is nu dus ook beschikbaar. Ik kan nu vanuit de database een deel van de back end maken (pijl linksonder), dit beschrijf ik later in hoofdstuk *pilot 3, deel 1*.  
Ik kan ook vanuit het systeemconcept een deel van de front end maken, de pijl rechtsboven:

Allereerst nemen we de te maken pagina erbij uit het systeemconcept, bijvoorbeeld een wedstrijdpagina. Deze hele pagina is opgebouwd uit de eerder beschreven design patterns. Voor alle design patterns zijn Smarty tags. Smarty is een abstractielaag die tussen PHP en HTML in zit. Met deze techniek is het mogelijk de PHP van de HTML te scheiden, wat betere code oplevert. Hierdoor wordt voldaan aan de kwaliteitseisen en onderhoudbaarheid die IAD voorschrijft.

Een voorbeeld van hoe een Smarty tag gebruikt wordt om de een design pattern weer te geven. We pakken een deel van een itempagina erbij:

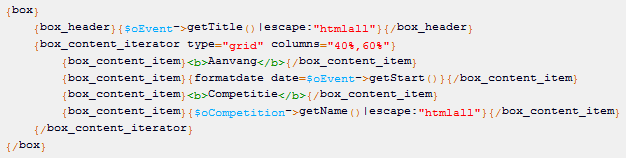
I**nzet: blok op itempagina.**



Dit blok is opgebouwd uit een bepaald type ‘box’, met een titel en een tabel met informatie. Er is in een documentatiepagina op Hyves te zien welke Smarty tags hiervoor gebruikt kunnen worden. Om dit blokje weer te geven is de volgende Smarty code nodig:

**Inzet: Snippet van Smarty code.**

De {box} {/box} code wordt door de server geïnterpreteerd en vervangt het voor de bijbehorende HTML code voor een blok. Zo worden alle Smarty tags behandeld en vervangen door de bijbehorende code. In het voorbeeld zijn ook variabelen te zien, bijvoorbeeld **$oEvent**. Dit is een variabele die PHP aan Smarty geeft, en staat voor een wedstrijdobject uit de database. Daarover meer in het hoofdstuk *Pilot 3, deel 1*.



De leden van het A-team (de ontwikkelaars) zullen de schermen uit het systeemconcept op de beschreven manier omzetten naar Smarty templates.

#### Activiteit 2: Bouw software bouweenheden

Het doel van deze activiteit is het daadwerkelijk omzetten van de mockups uit het systeemconcept naar Smarty templates, zoals beschreven. Ik heb in samenwerking met de opdrachtgever gekozen om met een workshop te beginnen. Deze workshop vindt plaats gedurende 1 dag.

##### Workshop / Hackaton

Om het programmeerwerk van het systeem een kickstart te geven organiseer ik een hackaton. Tijdens deze hackaton, waarbij 13 werknemers van Hyves aanwezig zullen zijn, werken we aan zowel pilot 2 als 3.   
Ik geef tijdens de workshop eerst een korte presentatie over het systeemconcept en bepaal daarna welke mensen aan welke onderwerpen zullen gaan werken. De onderwerpen voor de workshop staan in het PVA.

De workshop begon om 11:00 ’s ochtends en was uiteindelijk pas klaar om 23:00. Dat geeft wel aan dat er aardig wat werk is verzet.



*Na de workshop*  
Dankzij de workshop is er nu een werkend prototype. Dat wil zeggen dat de verschillende gebruikersschermen met basisfunctionaliteit werkend zijn en dat via een gebruikersaccount voorspellingen gedaan kunnen worden.



**Inzet: het overzichtsscherm van mockup naar werkend prototype.**

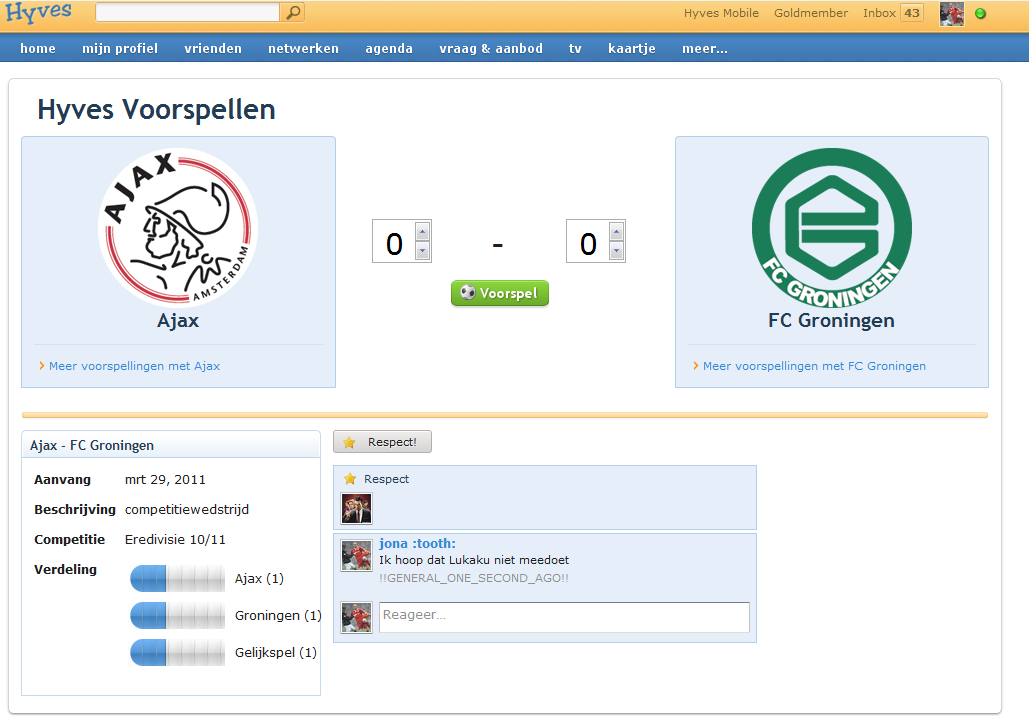
Zoals te zien is in dit figuur is dat de badges plaats hebben gemaakt voor punten. Het is veel werk om badges goed te maken. Het werken met punten is een stuk makkelijker. Voorlopig ga ik met Hyves Voorspellen uit van een puntentelling. De badges kunnen eventueel in een later iteratie weer terugkomen. Het overzichtsscherm is verder bijna gelijk aan de mockup uit het systeemconcept.



**Inzet: Het blok op de Hyves Homepage na de hackaton**

Het blok op de Hyves Homepage is qua functionaliteit werkend, maar heeft nog wat visueel opmaakwerk nodig. Dit neem ik op in het plan van aanpak.

**Inzet: De wedstrijdpagina na de hackaton.**



De wedstrijdpagina lijkt al erg veel op het systeemconcept. Er voor de verdeling gekozen voor een percentagebalk. Het wordt nog de vraag of dit zo blijft of dat er voor een andere weergave wordt gekozen. De Tell a Friend functie is ook nog niet opgenomen op deze pagina.

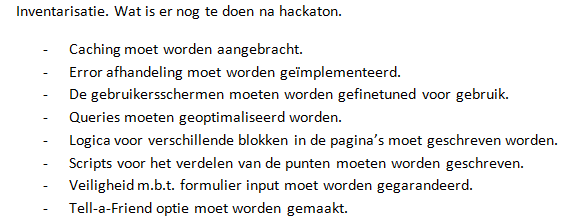
Een andere ontwikkeling die uit de workshop is gekomen is het signaal. Een signaal is een berichtje van een actie dat gedistribueerd wordt naar de vrienden van degene die de actie onderneemt. Voor Hyves Voorspellen wordt er een signaal gestuurd als iemand een voorspelling heeft gedaan. Dit zorgt voor het virale aspect. Vrienden van de voorspeller zien deze voorspelling en moeten daardoor worden aangespoord om ook een voorspelling te doen.



**Inzet: Signaal van een voorspelling.**

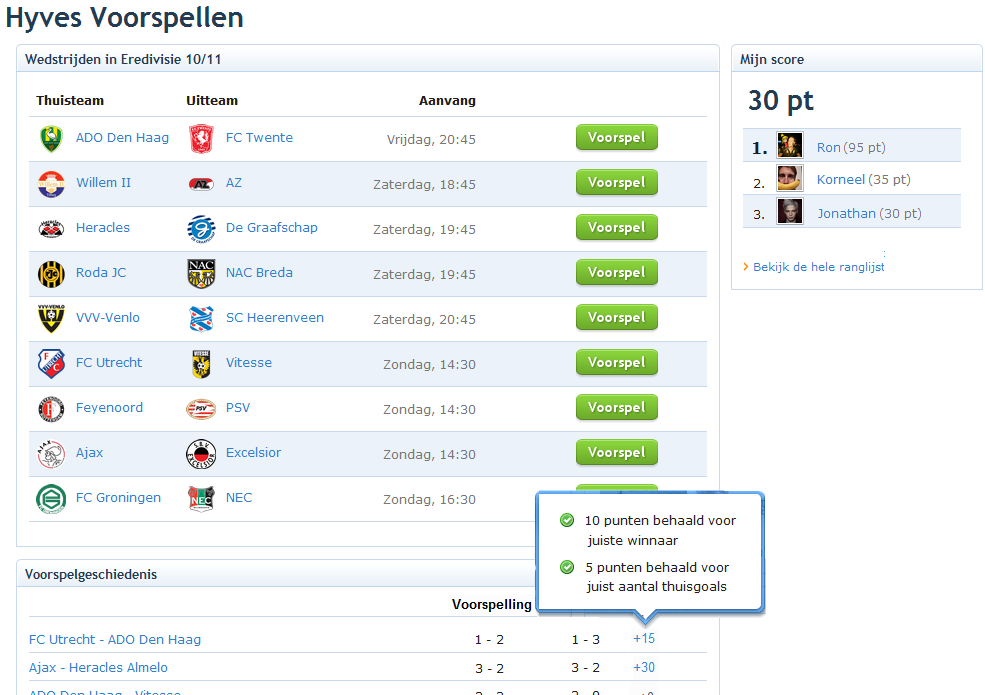
Na de hackaton maak ik een inventarisatie van wat er allemaal is gedaan en bespreek dit met de opdrachtgever. In deze gesprekken zijn we tot overeenstemming gekomen wat er nog moet gebeuren voordat we de eerste testfase in kunnen gaan. Ik heb deze punten opgenomen in het PVA.  
Ik ben me ervan bewust dat deze punten deels bij pilot 3 horen.

**Inzet: Deel uit te behandelen onderwerpen na workshop. Uit PVA**



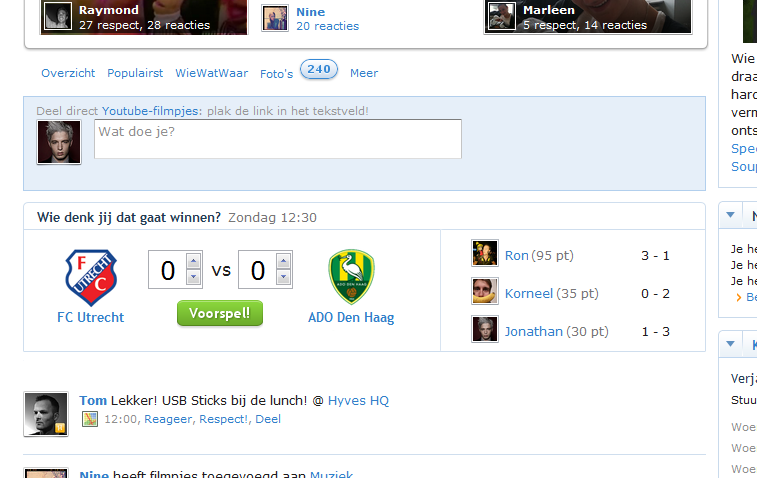
Deze punten zou ik in principe nu zelf moeten ontwikkelen, maar om een ander te versnellen heeft de opdrachtgever de hulp ingeschakeld van een collega. Deze collega is Kilian Marjew en is software engineer op Hyves. Ik geef aan wat er gedaan moet worden aan pilot 2 en 3, en samen ontwikkelen we dat.

Na twee weken zijn we klaar om het prototype te presenteren op Hyves. De aanpassingen die we gedaan hebben na de workshop leveren een prototype op dat in gebruik kan worden genomen door de testgroep.  
**Inzet: Hyves Voorspellen homepage.**



De Hyves Voorspellen homepage heeft enkele belangrijke veranderingen ondergaan:

- Om het duidelijk te maken dat het hier niet slechts om een programma gaat, maar dat het een navigatiepagina is, heb ik duidelijke CTA knoppen geplaatst naast de wedstrijden.  
- De lopende voorspellingen zijn van een blok uit de rechterkant naar de linkerkant verhuisd. Er is hier meer ruimte om ook de voorspelgeschiedenis hierin weer te geven. Ook worden het aantal behaalde punten voor die wedstrijd weergegeven. Als hier met de muis overheen wordt bewogen, wordt in een popup aangegeven hoe die punten verdiend zijn. Door met een popup te werken hoeft die informatie niet standaard weergegeven te worden, wat ervoor zorgt dat de pagina overzichtelijk blijft. (minimalistic design heuristic)  
- In het blok met de score is nu ook een ranglijst opgenomen. In deze ranglijst is te zien hoe een voorspeller het doet ten opzichte van zijn vrienden. Dit zorgt voor competitiviteit en zorgt er als het goed is voor dat gebruikers terugkomen om voorspellingen te doen.  
In dit blok staat ook een link om de gehele ranglijst te kunnen bekijken. De ranglijst op de Hyves Voorspellen homepage toont namelijk wegens ruimtegebrek enkel een top 5.  
- In samenwerking met de opdrachtgever is besloten om alleen op de Eredivisie te focussen. Dit komt vanuit de visie dat Hyves een Nederlands bedrijf is. Het competitieblok verdwijnt hierdoor.

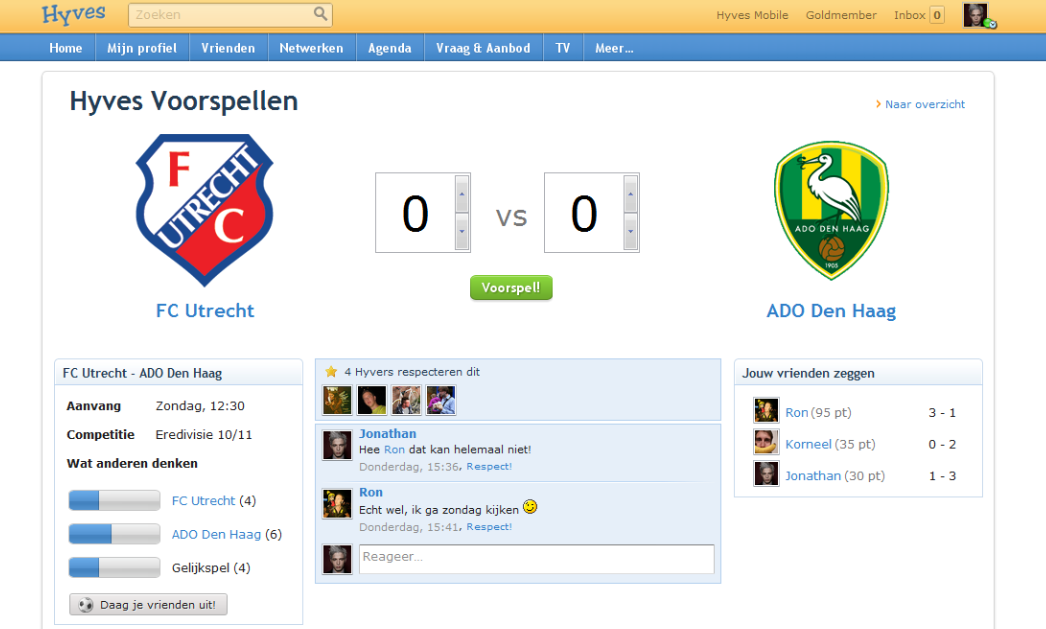


**Inzet: blok op de Hyves homepage**

Het blok op de homepage van Hyves heeft ook enkele belangrijke veranderingen ondergaan:

- Het is in een standaard box gezet, voor consistentie.  
- De laatste 3 vrienden die een voorspelling voor een deze wedstrijd hebben zijn hier te zien, samen met hun voorspelling. Dit is ook om de gebruiker aan te moedigen te voorspellen.  
- Er is gebruik gemaakt van het input element type ‘number’. Hierdoor kunnen de pijltjes gebruikt worden om een nummer in te vullen. Dit is een accelerator in het systeem, waardoor het systeem sneller kan worden bediend.

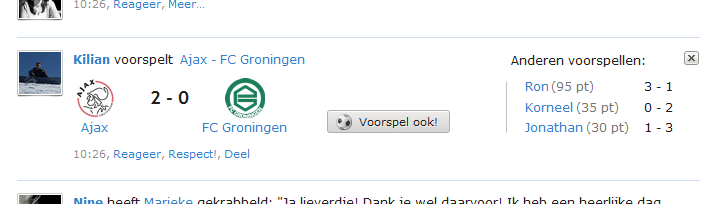
I**nzet: Wedstrijd pagina.**



De wedstrijdpagina is vrijwel gelijk gebleven aan de mockup uit het systeemconcept:

- De Tell a Friend functionaliteit is toegevoegd in de vorm van een knop aan het informatieblok. De knop opent een popup met de Tell a Friend functionaliteit. Dit is een standaard module op Hyves die hergebruikt is.  
- De percentagebalkjes in het informatieblok zijn netjes opgemaakt.  
- Aan de rechterkant is ook hier te zien welke vrienden ook deze wedstrijd hebben voorspeld.  
- De blauwe vlakken waar de logo’s in stonden zijn weggehaald. Deze keuze heb ik genomen omdat de vlakken, het design pattern meta-box, eigenlijk een afscheiding zijn voor meta-informatie. Het pattern werd dus niet juist gebruikt.  
- De oranje scheidingslijn is weggehaald. In het licht van minimalistic design, een van de usability heuristics, was deze scheidingslijn overbodig. De uitlijning van de bovenkant van de drie blokken vormt al een afscheiding.  
- Ook hier is gebruik gemaakt van het input element type ‘number’ als accelerator.

**Inzet: het signaal op de Hyves Homepage.**



De volgende aanpassingen zijn gemaakt aan het signaal:

- Er worden logo’s gebruikt om het visueel aantrekkelijker te maken. De logo’s kunnen ook herkend worden door de gebruiker, waardoor de aandacht wordt getrokken van de gebruiker.  
- Een CTA knop is geplaatst, zodat de aandacht van de gebruiker wordt gebruikt om hem aan te sporen om ook een voorspelling te doen.  
- Ook hier zijn de vrienden te zien die ook een voorspelling hebben gedaan voor deze voorspelling.

Pilot 1, deel 1 is nu afgerond en klaar om ingevoerd te worden voor de testgroep.

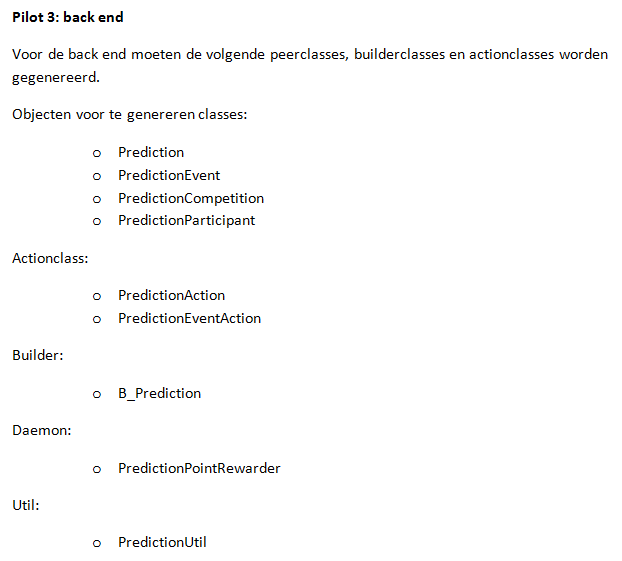
### Pilot 2: Back end, deel 1

De derde pilot die ik doorloop is het ontwikkelen van de back end voor Hyves Voorspellen.

De back end is het gedeelte van een systeem, waar de business logic staat. Het is code op een server die er bijvoorbeeld zorgt dat als een bepaalde URL wordt aangeroepen, de juiste pagina wordt weergegeven. Maar de back end doet meer dan alleen pagina’s weergeven. Als een webformulier met de score van een wedstrijd naar de server wordt verstuurd, zorgt de back end ervoor dat het wordt opgeslagen in de database en terugkoppelt aan de gebruiker of dat is gelukt.   
De back end is het technische gedeelte van het systeem ‘onder de motorkap’.

#### Activiteit 1: Plan van aanpak

Het doel van deze pilot is het ontwikkelen van de back end. De back end op Hyves bestaat uit een bepaalde structuur. In het plan van aanpak neem ik op welke onderdelen voor de back end moeten worden gemaakt. Vervolgens beschrijf ik in de activiteit *ontwerp en bouw software bouweenheden* waarvoor die onderdelen zijn en hoe ik ze maak. Omdat ik het proces het best uit kan leggen door direct voorbeelden en code weer te geven, en omdat het geen heel nieuw ontwerp is, heb ik het ontwerp en de bouw in 1 activiteit gedaan.

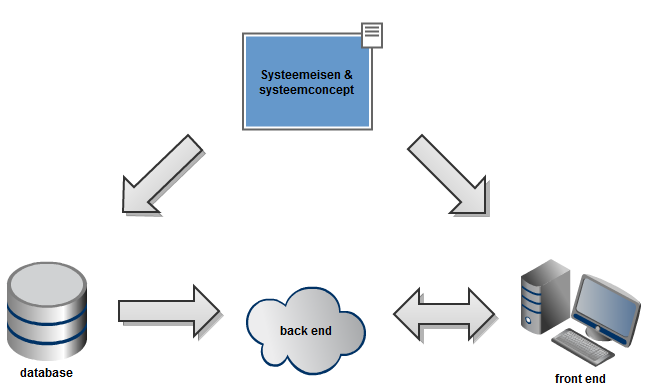


**Inzet: te ontwikkelen onderdelen voor de back end uit plan van aanpak**

#### Activiteit 2: Ontwerp en bouw software-bouweenheden

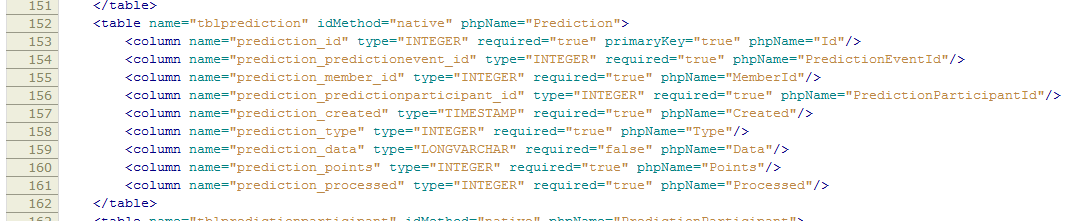
Het doel van deze activiteit is het specificeren van de technische structuur van de software-bouweenheden die deel uitmaken van deze pilot. De techniek die ik hierbij gebruik is procesbeschrijving van het genereren/aanmaken van de benodigde onderdelen. De techniek om de functionaliteit aan te brengen is *gestructureerd programmeren*.

Ik wil nogmaals kijken naar het figuur ‘volgorde ontwikkeling van pilots’ kijken.



**Inzet: volgorde ontwikkeling van pilots.**

Er staan vier objecten in het datamodel. Voor objecten wil ik PHP Classes genereren, zodat ze gebruikt kunnen worden in het prototype. Bij Hyves wordt gewerkt met een bepaald systeem, Propel, dat de communicatie tussen database en PHP verzorgt. Dit programma heeft input nodig in een bepaald formaat om de scripts te kunnen genereren die deze communicatie verzorgt. De eerste stap is dus niets meer dan het invoeren van het datamodel in een bepaalde XML-structuur.



**Inzet: XML structuur voor Propel.**

Propel gebruikt het Xml-bestand als input en genereert aan de hand daarvan een aantal PHP-Classes voor elk object:

* + Base Class:   
    Wordt bij elke aanpassing van het datamodel opnieuw gegenereerd.   
    Het is het skelet van elk object en bevat alle properties.
  + Object Class:   
    Wordt eenmalig gegenereerd.   
    Extends Base Class.

Bevat functionaliteit die op het object van toepassing zijn.

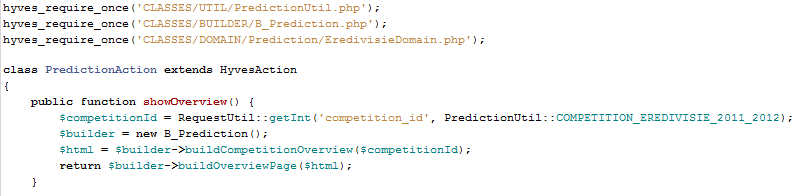
* + BasePeer Class:  
    Wordt bij elke aanpassing opnieuw gegenereerd.  
    Bevat informatie en functionaliteit voor de connectie met de database.
  + Object Peer Class:  
    Wordt eenmalig gegenereerd.  
    Extends BasePeer Class.

Bevat op maat gemaakte functionaliteit voor het ophalen van objecten.

Deze classes zijn een belangrijk deel van de back end. Het genereren van de classes is de pijl linksonder uit het figuur ‘volgorde ontwikkeling van pilots’. De gegenereerde bestanden vormen een beginpunt waarmee gewerkt kan worden. Het is nu mogelijk te werken met de objecten en ze op te slaan of op te halen.

Ik heb nu alleen de pijl rechtsonder nog niet beschreven. De dubbele pijl betekent een wisselwerking tussen de front end en de back end. Ze worden tegelijkertijd ontwikkeld. Een deel van de back end wordt gevormd door de front end en andersom. Ik zal dit nader verklaren.  
  
*Action classes*  
Een action class op Hyves staat voor een bepaalde pagina. Het is 1 ding dat het systeem om kan gaan met de objecten, het moet vervolgens ook weten wanneer het iets moet doen en wanneer het iets moet tonen. Om het systeem dit te laten weten gebruikt Hyves ‘modules’ en ‘actions’. Modules zijn een bepaald onderdeel van het Hyves systeem en actions zijn bepaalde taken die in die modules zijn gedefinieerd. Dit komt als volgt ten uiting:

Een URL op Hyves kan zijn: <http://hyves.nl/?module=prediction&action=showOverview>. Het systeem weet nu dat het de *Prediction* module (genaamd predictionAction) moet gebruiken en de *showOverview* functie uit moet voeren. In die functie schrijf ik de logica die de overzichtspagina toont.



**Inzet: Snippet uit predictionAction class.**

Omdat deze URL niet zo mooi is en niet zoekmachine-vriendelijk voeg ik nog een bepaalde mapping aan het systeem toe. <http://hyves.nl/voetbal> brengt je ook naar de overzichtspagina.

We hebben voor Hyves voorspellen twee nieuwe pagina’s gemaakt in de front end, de Hyves Voorspellen homepage en de wedstrijdpagina. We moeten dus twee action classes aanmaken in de back end.

In het kleine stukje van de predictionAction is te zien dat een *prediction builder* (genaamd B\_Prediction) wordt aangeroepen. Deze builder classes bevatten de code die de een bepaald deel van de pagina in html teruggeven. Zo kunnen deze blokken hergebruikt worden op andere pagina’s. Een eis die IAD aan de software stelt. Op deze manier wordt de front end ook gevormd door hoe de back end eruit ziet.



I**nzet: Snippet uit de B\_Prediction class.**

In de B\_prediction class is te zien dat een template (overview.tpl) wordt aangeroepen. Dit is een van de front end Smarty templates die in pilot 2, deel 1 zijn gemaakt. Bepaalde variabelen worden in deze builder class aan de templates toegewezen, zodat deze kunnen werken met de benodigde data.

Een andere class die te zien is in het stukje code is de PredictionUtil. Dit is een op zichzelf staande class met helpfuncties. In het voorbeeld wordt de functie canUsePredictions aangeroepen. Deze controleert of de databaseservers bereikbaar zijn. Mocht dat niet zo zijn wordt er een foutmelding getoond (Error prevention). Ook staat er in deze functie een check op de AppState. De AppState is een soort aan- en uitschakelaar, waarmee functies op Hyves aan en uit kunnen worden gezet. Het is hiermee ook mogelijk de applicatie alleen aan te zetten voor een testgroep. Op deze manier maken we vanuit de database en de front end de back end. We zorgen ervoor dat alle functionaliteit uit het systeemconcept werkt.

*Daemon*  
Als laatst is er in het plan van aanpak een onderdeel te zien dat PredictionPointRewarder heet. Dit is een zogenoemde daemon. Een daemon is een scriptje dat continue wordt uitgevoerd. In dit geval bekijkt het scriptje continue of er wedstrijden zijn voorspeld en of de uitslag daarvan al beschikbaar is. Als dat zo is zal het scriptje een bepaald aantal punten toewijzen aan de voorspelling van de gebruiker.

#### Activiteit 3: Test en evalueer pilotdeel

Het doel van deze activiteit is het beoordelen en testen van een of meer delen van de pilot. De opdrachtgever zit in de kamer tegenover mij en we hebben regelmatig een meeting met een interaction designer en mijn stagebegeleider erbij. Het evalueren van de verschillende pilots en het hele prototype gebeurt dus niet in 1 workshop maar gedurende verschillende momenten in het ontwikkelproces.

In een dergelijke meeting bespreken we – eerst aan de hand van mockups, later het echte prototype – of alles naar idee wordt uitgevoerd en wat er beter kan of kan worden toegevoegd. De systeemeisen worden dus regelmatig herzien, wat prima kan in de gekozen strategie.

Om continue werking van de producten op Hyves te kunnen garanderen, worden automatische regressietests gemaakt. Deze tests maken gebruik van een Pythonlibrary, genaamd Selenium. Het bootst een gebruiker na die bepaalde acties op een webpagina uitvoert. Ik schrijf een test voor de functies uit de taakdiagrammen.



**Inzet: Stuk code uit de regressietest**

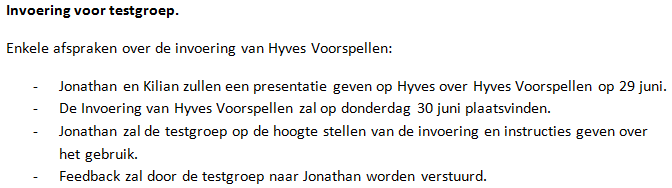
### Invoering voor testgroep

Het doel van deze fase is het operationeel maken van de pilot in de gebruikersomgeving. Het prototype is nu werkend en klaar om ingevoerd te worden. Het prototype is met grote zorg ontwikkeld, maar nu is het belangrijk om te zien wat de ervaring van gebruikers is. De testgroep bestaat uit een groep van ongeveer 150 gebruikers van Hyves.

#### Activiteit 1: Plan van aanpak

Het doel van deze activiteit is afspraken maken over het daadwerkelijke invoeren van het prototype. In het plan van aanpak heb ik enkele afspraken die voor deze invoering belangrijk zijn opgenomen.

**Inzet: deel uit plan van aanpak over invoering**



De invoering zelf is niet erg ingewikkeld. Het prototype wordt met de minor release van 30 juni verspreid over de webservers. Dit wordt zoals bij elke release gedaan door degene die de release uitvoert. Als dit gedaan is zal ik de AppState aanzetten, zodat Hyves Voorspellen aan staat voor de testgroep. Hyves Voorspellen wordt dan aan de testgroep aangeboden. Zij hoeven dus niet per se mee te doen aan Hyves Voorspellen, wat een idee geeft over de doelgroep.

#### Activiteit 2: Voer pilots in

Het doel van deze activiteit is het invoeren van de pilots in de breedste zin van het woord. Voorafgaand aan het daadwerkelijke invoeren van de pilots geef ik samen met de collega waarmee ik na de hackaton aan het prototype heb gewerkt een presentatie. Een dergelijke presentatie wordt op Hyves altijd gegeven als er een nieuw product wordt gelanceerd. Het doel hiervan is het op de hoogte brengen van stakeholders en andere geïnteresseerden in het bedrijf.   
Het is voor verschillende mensen in het bedrijf belangrijk om op de hoogte te zijn van nieuwe producten:

* + Community: Voor vragen over Hyves worden gebruikers naar de Helpdesk gestuurd. De medewerkers van deze helpdesk moeten dus weten hoe een product werkt.
  + Marketing: Marketing wil altijd graag weten of er mogelijkheden zijn om geld aan een nieuw product is te verdienen.
  + De opdrachtgever / directie. Uiteraard is de opdrachtgever aanwezig.

Er is bij de presentatie een opkomst van ongeveer 50 personen. De presentatie is gefocust op de functionaliteit van Hyves Voorspellen en niet zozeer op hoe het product tot stand is gekomen. Het prototype is bij de demonstratie goed ontvangen, wat belangrijk is voor de pilotacceptatie.  
De dag na de presentatie is Hyves Voorspellen ingevoerd voor de testgroep.



**Inzet: Demonstratie van Hyves Voorspellen.**

### Pilot 1,2 en 3: deel 2 (bugfixes)

Zoals we hebben gezien in het ontwikkelscenario is de tweede iteratie van alle drie de pilots bestemd voor bugfixes. Een bug is een fout in de software, waardoor een systeem niet, of niet helemaal volgens de specificaties werkt.

#### Activiteit 1: Verzamel bugs en los ze op

Het doel van deze activiteit is het verzamelen van de bugs en deze op te lossen. Vervolgens moet er ook gezorgd worden dat dezelfde fout niet nog een keer voor kan komen. Dit wordt gedaan door de automatische test aan te passen, zodat deze bij eenzelfde fout alarm slaat.

Het verzamelen van de fouten gebeurt aan de ene kant vanzelf, aangezien de testgroep mij mailt als ze een fout tegenkomen. Aan de andere kant controleer ik de database op foutieve data. Ik doe dit ook door de business rules in m’n achterhoofd te houden. Iemand moet bijvoorbeeld geen twee voorspellingen voor één wedstrijd kunnen hebben.

Ik ben tijdens deze iteratie enkele softwarefouten tegengekomen. Zo was het mogelijk om een kommagetal in te vullen bij een voorspelling. Ook resulteerde het aanpassen van een voorspelling soms in een dubbele voorspelling.

Omdat het niet belangrijk is voor het onderzoek, zal ik hier niet verder op ingaan. Het is natuurlijk wel cruciaal dat de problemen worden opgelost.

### Invoering

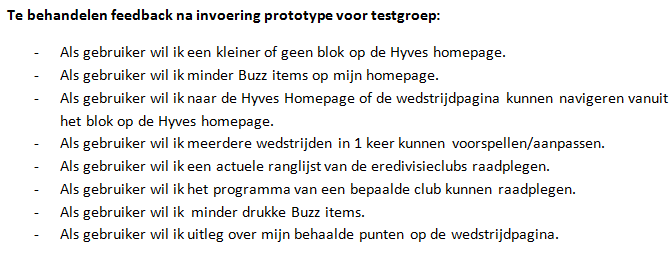
Normaal gesproken worden bugfixes direct nadat ze zijn opgelost met een *stable release* live gezet. Omdat Hyves Voorspellen nog in de testfase zit en omdat een release altijd risico’s met zich meebrengt, worden de bugfixes gepland voor de *minor release* van de week erna.

### Definitiestudie 2

De tweede definitiestudie doe ik naar aanleiding van de feedback van de testgroep. De manier van feedback verzamelen noemt de IAD methode ‘met behulp van operationele prototyping’. Testpersonen gebruiken een prototype en geven aan de hand daarvan feedback. Aan de hand van deze feedback wordt het prototype aangepast. Deze en de volgende twee iteraties worden volgens deze methode uitgevoerd.

#### Activiteit 1: Stel pilotplan op

Het doel van deze activiteit is het organiseren van de feedback, en het plan voor de komende iteratie maken. Ik verzamel alle feedback en overleg deze met de opdrachtgever. De aanbevelingen die we willen gaan invoeren, neem ik op in het plan van aanpak in de vorm van user stories. Op deze manier is het duidelijk wat de gebruiker precies wil. Dit doe ik gedurende deze en de volgende iteratie. Mocht ik nog feedback krijgen tijdens de laatste iteratie, zal ik deze meegeven aan degene aan wie ik het product overdraag.



**Inzet: User stories van de feedback die is verzameld.**

De User stories staan gerangschikt op hoe vaak dezelfde feedback werd gegeven. De meeste meldingen gingen over het blok op de homepage en de buzz items. Daarom heb ik besloten deze pilot deze twee onderdelen ga aanpassen.

### Pilot 2 en 3: deel 3

#### Activiteit 1: Ontwerp en bouw software-bouweenheden

Het doel van deze activiteit is het ontwerpen en implementeren van een oplossing voor de wens van de gebruikers. De technieken die ik hiervoor gebruik zijn gestructureerd programmeren en Interface Design.

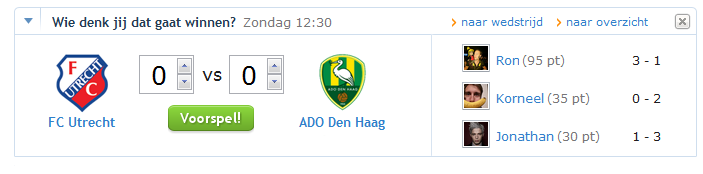
Het eerste probleem is dat het blok op de homepage te groot is, of überhaupt aanwezig is. Dit zijn eigenlijk twee problemen en ik ga voor beide problemen proberen een oplossing te vinden. Het volgende probleem is dat gebruikers niet vanuit het blok naar een volgende pagina kunnen navigeren. Dit is een makkelijker op te lossen probleem.

Om het blok op de homepage kleiner te maken is er een design pattern beschikbaar op Hyves, een zogenaamde collapseable box. Het blok klapt dan als het ware in, en alleen de header is dan nog zichtbaar. Dit lijkt me een goede oplossing voor het eerste probleem.

Het maken van een sluitknop, die de gehele box weghaalt, zou in de andere wens van de gebruiker kunnen voorzien. Om te zorgen dat de box gesloten of ingeklapt blijft, sla ik per gebruiker in Memcached op wat de status van de box is. Dit cachesysteem is als het ware een tabel met 2 kolommen: een key en een value. De key is in dit geval “PREDICTION\_HOME\_BOX\_STATE \_MEMBER\_XXX”, de value is de state (open, ingeklapt of gesloten). “XXX” wordt vervangen door het unieke gebruikers id. Dit cachesysteem heeft als voordeel dat er geen nieuwe database hoeft te worden aangemaakt. Het is ook erg snel op te halen, omdat het in het RAM geheugen van een server wordt opgeslagen.

Als er een week later weer een nieuwe wedstrijd is willen we de gebruiker weer aanmoedigen om te voorspellen. Daarom verwijderen we dan de cache. De box wordt dan weer gewoon normaal getoond met de nieuwe wedstrijd.

Voor de user story ‘Als gebruiker wil ik naar de Hyves Homepage of de wedstrijdpagina kunnen navigeren vanuit het blok op de Hyves homepage.’ voeg ik twee links toe aan de header van het blok.



**Inzet: Het blok op de homepage na aanpassingen.**

Het volgende onderdeel dat ik in deze iteratie behandel is het buzz item. De testgebruikers melden dat er teveel buzz items op hun homepage verschijnen. Nu wordt er met elke voorspelling die gedaan wordt een buzz item verspreid naar de vrienden van die gebruiker. Dat kunnen dus 9 items zijn per speelronde, dus per week. Omdat een gebruiker veelal meerder wedstrijden achter elkaar zal voorspellen, worden deze items vlak na elkaar verstuurd. Om dit probleem te verhelpen wordt een nieuwe regel dat er slechts 1 buzz item wordt verstuurd per persoon per ronde.

Gebruikers uit de testgroep meldden ook dat het buzz item veel drukker/voller is dan andere updates op de homepage. Buzz item wat rustiger te maken, haal ik het lijstje van vrienden die ook deze wedstrijd hebben voorspeld uit het buzz item. Dit is consistent met de interaction design heuristic ‘minimalist design’, die voorschrijft geen onnodige informatie te tonen.



**Inzet: buzz item na aanpassing.**

### Invoering

De aanpassingen uit deze iteratie zijn direct met de minor release live gezet. Dit is een week nadat de bugfixes zijn gereleased.

### Definitiestudie 3

De derde definitiestudie is een zeer korte definitiestudie. Er is geen nieuwe feedback gekomen van de testgroep die geïmplementeerd dient te worden. Er is wel meer van dezelfde feedback gekomen. Ik pak de userstories op die het hoogst op de lijst staan.

#### Activiteit 1: Stel pilotplan op

Het doel van deze activiteit is beslissen welke onderdelen ik ga aanpassen naar aanleiding van de feedback van de gebruikersgroep.

In deze pilot zal ik mij bezig houden met twee userstories:

* + Als gebruiker wil ik meerdere wedstrijden in 1 keer kunnen voorspellen/aanpassen.
  + Als gebruiker wil ik een actuele ranglijst van de eredivisieclubs raadplegen. Ik kies

Ik kies deze twee omdat ze qua urgentie bovenaan staan en omdat ze alle twee van toepassing zijn op de Hyves Voorspellen homepage.

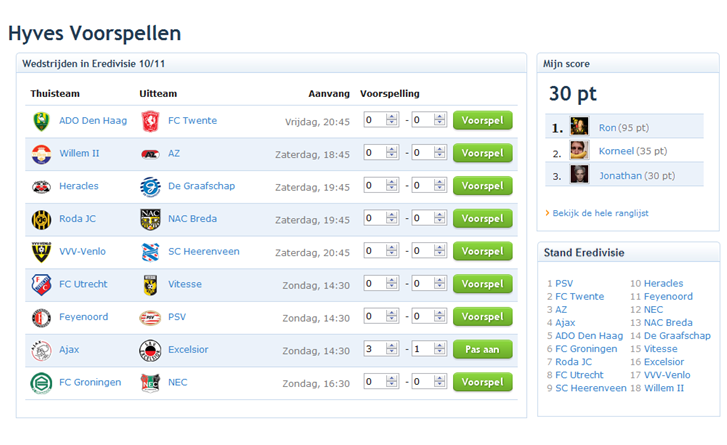
### Pilot 2 en 3: deel 4

#### Activiteit 1: Ontwerp en bouw software-bouweenheden

Het doel van deze activiteit is het ontwerpen en implementeren van een oplossing voor de wens van de gebruikers. De technieken die ik hiervoor gebruik zijn gestructureerd programmeren en Interface Design.

Om meerdere wedstrijden tegelijk te kunnen voorspellen, moet de navigatiepagina, de Hyves Voorspellen Homepage, meer dan alleen een navigatiepagina worden. De gebruiker moet hier direct zijn voorspellingen in kunnen vullen voor alle te voorspellen wedstrijden. De links moeten dus webformulieren worden. Vervolgens moet de back end aangepast worden om hiermee om te kunnen gaan.

De volgende userstory bestaat uit een wens voor nieuwe functionaliteit. Gebruikers willen de stand van de eredivisie in kunnen zien. Waarschijnlijk kent niet iedereen alle clubs even goed, en helpt dit ze bij het maken van een voorspelling. Toevallig is er door het wegvallen van het blokje ‘Competities’ wat ruimte vrijgekomen voor een ander blok. Ik gebruik deze ruimte dan ook voor een blok met de stand. De informatie voor deze stand wordt aangeboden via een API. Omdat deze informatie slechts 1 keer per week veranderd, is het wat overdreven is om voor deze informatie het datamodel voor aan te passen. In plaats daarvan laat ik de daemon deze informatie één keer per week ophalen en in Memcached zetten.



**Inzet: Links het blok met wedstrijden op de Hyves Homepage. Rechtsonder het blok met de stand.**

### Invoering

De aanpassingen uit deze iteratie zijn direct met de minor release live gezet. Dit is een week na de invoering van de vorige iteratie.

### Definitiestudie 4

De vierde definitiestudie is een korte definitiestudie. Dit is de laatste definitiestudie in het ontwikkelscenario. Het is niet verstandig om nu nog grote aanpassingen aan het systeem te doen, aangezien het over een week wordt geopend voor heel Nederland. Er zijn veel ogen gericht op het systeem en het moet goed werken.

Er is nog één userstory die niet behandeld is:

* + Als gebruiker wil ik uitleg over mijn behaalde punten op de wedstrijdpagina.

Deze userstory zal ik in de laatste iteratie behandelen.

### Pilot 2 en 3: deel 5

#### Activiteit 1: Ontwerp en bouw software-bouweenheden

Het doel van deze activiteit is het ontwerpen en implementeren van een oplossing voor de wens van de gebruikers. De technieken die ik hiervoor gebruik zijn gestructureerd programmeren en Interface Design.

Deze wens van de gebruikers is eigenlijk het hergebruiken van eerder gemaakte functionaliteit. Het gaat hier om het tonen van de uitleg over behaalde punten na een wedstrijd. We hebben dit op een eerdere afbeelding van de Hyves Voorspellen homepage gezien.

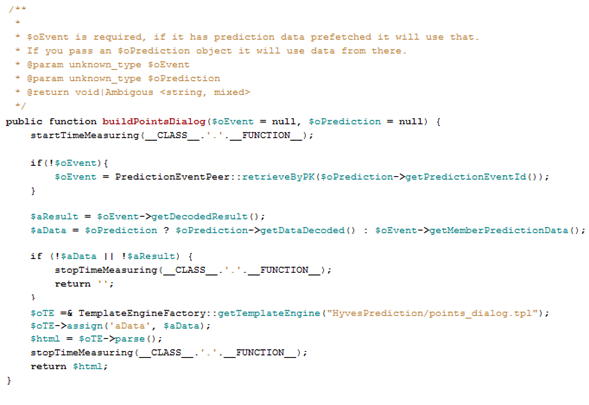


**Inzet: Uitleg over de behaalde punten op de Hyves Voorspellen Homepage.**

Doordat deze functionaliteit in een builder class is gezet, is het simpel om deze op een andere plek ook weer te geven. Ook hier wordt, bij een muisklik of als de muis over het aantal punten wordt bewogen, een popup getoond met uitleg over de behaalde punten.

I**nzet: Een deel van de wedstrijdpagina als een wedstrijd is afgelopen.**





**Inzet: Het deel uit de builder class dat de popup weergeeft. Herbruikbaarheid is belangrijk.**

Dit is de laatste functionele aanpassing aan het Hyves Voorspellen prototype. Het is nu geen prototype meer, maar een product dat klaar is om in gebruik te nemen.

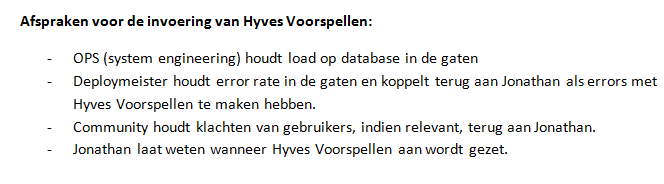
### Fase 3: Invoering

Het doel van deze fase is het operationeel maken van de pilot in de gebruikersomgeving. Het daadwerkelijke invoeren van het prototype dus. De eerste stappen in dit proces zijn het maken van een plan van aanpak en een meetplan.

#### Activiteit 1: Plan van aanpak

Om de invoering van het project soepel te laten verlopen, moet er een plan van aanpak gemaakt worden. Het is belangrijk dat iedereen die iets met de invoering te maken heeft weet wanneer er iets gaat gebeuren. Ik denk hierbij aan degene die de nieuwe code gaat *deployen* over de webservers (de deploymeister), maar ook de system engineers die in de gaten moeten houden of de databases de nieuwe *load* aankunnen.

Er zijn twee grote voordelen. De medewerkers van Hyves hebben veel ervaring met het live zetten van nieuwe producten. Daarom weten de betrokken personen waar ze op moeten letten. Een ander voordeel is dat het product eigenlijk al bijna helemaal online staat. Bij de laatste release wordt alleen een kleine aanpassing meegenomen. Het belangrijkste verschil is het grote aantal gebruikers dat Hyves Voorspellen gaat gebruiken.



**Inzet: deel uit plan van aanpak met afspraken over de invoering.**

### Activiteit 2: Stel meetplan op

Het doel van deze activiteit is een plan op te stellen, om de juiste meetapparatuur aan te brengen. Het is belangrijk om te zien wat voor gedrag de gebruikers vertonen. Door bepaalde eigenschappen te meten kunnen later verbeteringen aangebracht worden aan het systeem.

Veel van deze informatie is uit de database te halen, omdat we het voorspellingsobject aan het memberobject hebben gekoppeld. Uiteindelijk moeten de volgende segmentatiecriteria per ronde worden gemeten om een goed beeld van de gebruikersgroep te krijgen:

* + Aantal deelnemers.
  + Aantal voorspellingen.
  + Gemiddeld aantal punten.
  + Gemiddelde leeftijd per ronde.
  + Leeftijdsgroep per ronde.
  + Geslacht per ronde.
  + Percentage nieuwe/terugkerende gebruikers.

Door naar de terugkerende gebruikers te kijken kan er gevraagd worden aan de gebruikers die niet terugkeren waarom ze dat doen. Dit valt buiten de scope van dit onderzoek, maar is wel belangrijk om te meten.

Ik breng de meetpunten aan in de code en zet ze klaar om mee te worden genomen met de aankomende release.

### Activiteit 3: Voer pilots in

Het doel van deze activiteit is het daadwerkelijke invoeren van het systeem. Nadat het de aanpassingen met de release meegenomen zijn, zet ik door middel van de AppState Hyves Voorspellen aan. Hyves Voorspellen is nu beschikbaar voor iedereen.

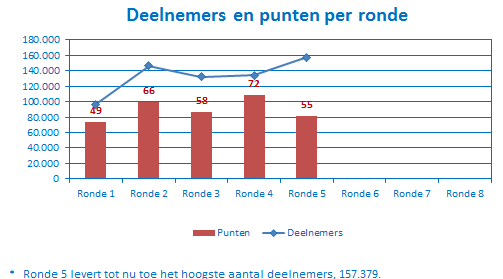
### Activiteit 4: Draag pilots over

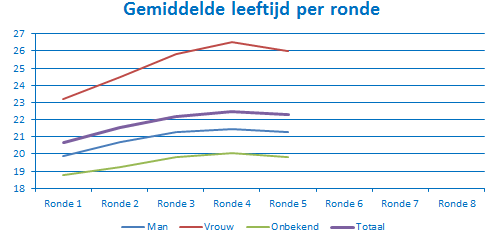
Na het invoeren van het product is moment daar om het over te dragen aan de nieuwe Product Owner. Dit doe ik nadat de opdrachtgever akkoord heeft gegeven over de overdracht en heeft bepaald dat hij tevreden is over de afgesproken resultaten.

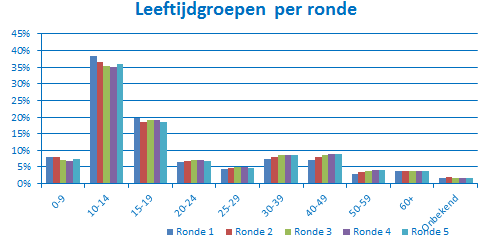
De Product Owner is Tim Besselink, hij is vanaf nu verantwoordelijk voor het systeem. De code, plaatjes, datamodel, et cetera zijn beschikbaar in het centrale versiecontrole systeem. Ik hoef deze dus niet fysiek over te dragen. De documentatie is beschikbaar op het intranet. Uiteraard blijf ik voorlopig beschikbaar voor ondersteuning en vragen over de werking van het systeem.

### Activiteit 5: Evalueer testresultaten

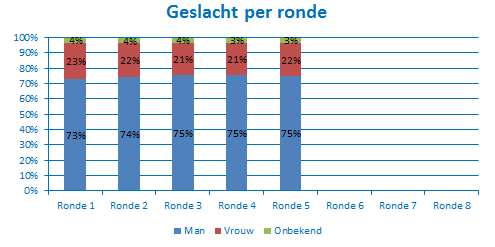
Deze activiteit behoort niet tot de ontwikkeling van het project. Ik neem deze activiteit alleen op om te laten zien hoe de eerder aangebrachte meetpunten kunnen worden weergegeven in een rapport. Hiernaast is het interessant om te zien hoe de gebruikersgroep van dit systeem eruit ziet. Ik heb dit rapport niet zelf gemaakt en zal daarom ook niet beschrijven hoe het tot stand is gekomen.

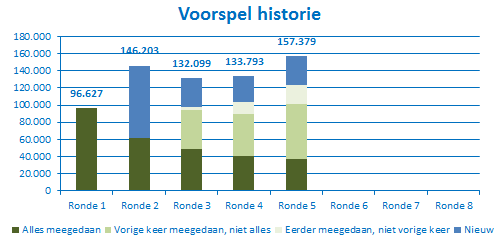


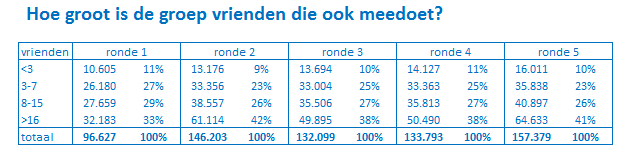




Inzet: De gebruikersgroep gevisualiseerd.







# 3. Deel III, conclusie

In deel drie van dit verslag zal ik terugkijken op het project. Eerst zal ik het proces evalueren. Daarna kijk ik terug op het product en of het geworden is wat ik aan het begin van het project voor ogen had.

## 3.1 Procesevaluatie

In dit hoofdstuk evalueer ik het ontwikkelingsproces. Ik kijk daarbij naar de technieken: Heb ik de juiste technieken gebruikt en heb ik deze op een juiste manier gebruikt? Ik deze vragen behandelen per fase, volgens het ontwikkelscenario.

### 3.1.1 Definitiestudie

Bij het ontwikkelen van Hyves Voorspellen heb ik veelvuldig gebruik gemaakt van het boek “Evolutionair ontwikkelen van informatiesystemen” van R.J.H. Tolido. Dit boek heeft een zeer uitgebreide handleiding voor het gebruiken van de IAD methode. De meeste technieken die ik heb worden in dit boek aangeraden. De onderdelen van de definitiestudie komen ook uit dit boek.

#### Stel plan van aanpak op

*Gebruikte technieken: Analyse van cruciale succesfactoren, Vastlegging organisatiestructuur en planningstechnieken.*Het plan van aanpak is vooral belangrijk om afspraken vast te leggen. Omdat dit de eerste keer was dat ik de IAD methode gebruikte en aangezien ik de definitiestudie voornamelijk alleen maakte, wist ik niet zo goed welke afspraken gemaakt moesten worden. Inmiddels weet dat beter. Het is bijvoorbeeld belangrijk om af te spreken in welk detail het systeemconcept wordt gemaakt.  
Mijn plan van aanpak is in dit project nogal summier en dit zou ik in de toekomst anders doen.   
Ik heb bijvoorbeeld gezien dat het gebruik van taakdiagrammen helpt het systeem te begrijpen voordat het systeem er is. Ik zou dan in een volgend plan van aanpak ook opnemen dat er taakdiagrammen gemaakt moeten worden. Dit geldt ook voor andere waardevolle technieken.

#### Definieer ontwikkelscenario

*Gebruikte technieken: workshop, iteratiestrategie visualisatie.*  
Ik heb deze activiteit initieel niet volledig begrepen. Ik dacht dat het puur ging om het verkrijgen van een beeld van wat het systeem moet kunnen en hoe het globaal moet werken. Ik heb hier zoals beschreven een *Request For Comments* voor aangemaakt, wat interessante inzichten opleverde. Na een gesprek met mijn scriptiebegeleider werd duidelijk dat deze activiteit ook belangrijk is om de iteratiestrategie te bepalen en visualiseren. Achteraf gezien is dit zeker een belangrijk onderdeel geweest en aan de hand hiervan de pilots kan plannen. Ik vind wel dat het iets te vroeg in de definitiestudie staat. Ik denk dat ik een beter scenario kan maken als ik de systeemeisen en liefst ook het systeemconcept heb. Ik denk dat ik de technieken goed heb gebruikt, aangezien het een helder scenario heeft opgeleverd.

#### Definieer systeemeisen

*Gebruikte technieken: workshop.*  
Het is niet zo dat de workshop die ik heb gedaan met de opdrachtgever direct een lijst met systeemeisen oplevert. Ik weet ook niet of dat kan. In mijn geval was het meer een bespreking van de wensen en ideeën die geuit zijn in het RFC. Het prioritiseren en filteren van deze wensen gaf een aantal punten die ik als basissysteemeisen heb gebruikt. Ik denk dat het heel belangrijk is om je hierbij voor ogen te houden welke ontwikkelmethode en strategie is gekozen. Omdat bij dit project de systeemeisen continue veranderen, is het geen probleem om zulke algemene systeemeisen te hebben. Als er voor een strategie is gekozen waarbij de systeemeisen van tevoren vaststaan, zijn deze systeemeisen absoluut niet toereikend. Die moeten dan veel gedetailleerder zijn en daarvoor zijn dan andere technieken voor nodig.   
Wat ik in de toekomst anders zou doen is dat ik de systeemeisen beter zou bijhouden. Ik heb nu niet steeds bij elke aanpassing aan het systeem de systeemeisen aangepast. Het heeft in dit project geen problemen opgeleverd, maar ik kan me voorstellen dat dit wel zou kunnen gebeuren.

#### Bepaal systeemconcept

*Gebruikte technieken: Taakdiagrammen, wireframes/mockups, flowbeschrijving.*  
Ik zie dit als een zeer belangrijk onderdeel van de definitiestudie. Ik, en anderen die aan het project hebben gewerkt, hebben vaak teruggekeken naar de producten van deze activiteit bij het ontwikkelen. De taakdiagrammen vertalen de systeemeisen goed naar de hoe het systeem moet werken, gezien vanuit het perspectief van de gebruiker. Ik heb de belangrijkste acties van de gebruiker uit de systeemeisen gebruikt om de taakdiagrammen te maken en ik denk dat dat goed heeft gewerkt. Ik ben tevreden met de taakdiagrammen die ik heb gemaakt.  
Aan de hand van die taakdiagrammen, en dan vooral vanuit het beeld van het systeem dat daardoor is ontstaan, kon ik de wireframes maken. Als ontwikkelaar krijg ik vaak de vraag of een bepaalde pagina gemaakt kan worden. Als daarbij dan een wireframe wordt geleverd ben ik daar erg blij mee: het betekent dat de opdrachtgever heeft nagedacht over de pagina en het geeft een duidelijk beeld van wat er gemaakt moet worden. Omdat er na het zien van de wireframes nog onduidelijkheden kunnen blijven bestaan over de werking van het systeem, heb ik er een begeleidend document voor geschreven. Al met al geven de ze 2 producten een goed beeld van het systeem. Dat bleek uit het feit dat de ontwikkelaars in de hackaton geen enkel probleem hadden met het ontwikkelen van het prototype aan de hand van de producten.

#### Beschouw technische structuur

*Gebruikte technieken: beschrijving techniek.*Aangezien ik al een tijd bij Hyves op de development afdeling werk, weet ik precies hoe de technische omgeving eruit ziet. Ik had dan ook geen probleem om het een en ander op papier te zetten over de technische structuur. Ik had daarentegen wel wat meer moeite met de juiste dingen op papier te zetten. Ik had eerst gekozen voor het beschrijven van onderdelen die belangrijk zijn voor Hyves. Na een aanwijzing van mijn scriptiebegeleider werd mij duidelijk dat dat niet de goede focus was. Het is natuurlijk alleen noemenswaardig in het beschouwen van de technische structuur als het belangrijk is voor dit project. Daarom heb ik aardig wat geschrapt uit dit hoofdstuk en in plaats daarvan geschreven over onderdelen van de technische omgeving waardoor ik bepaalde keuzes voor het project moest maken. Ik vond dit een erg goede aanwijzing en ben na aanpassing tevreden over het product van deze activiteit.

Beschouw organisatorische inrichting

*Gebruikte technieken: vastleggen van de veranderingen in organisatiestructuur*Ook dit onderdeel had ik niet in een keer goed. Ik schreef dat dit onderdeel niet nodig was omdat Hyves voorspellen een zelfonderhoudend systeem zou worden. Dit is in zekere zin ook zo, alleen het is natuurlijk niet zo dat het aan wordt gezet en er vervolgens nooit meer naar hoeft te worden gekeken. Wat vooral moet worden gedaan is dat er in de gaten moet worden gehouden dat het systeem blijft werken. Het is belangrijk om duidelijk afgesproken te hebben wie deze verantwoordelijkheid draagt. Er hoeft dus niet veel gewijzigd te worden aan de organisatorische inrichting, maar dat er helemaal niets verandert is ook niet waar.

#### Stel pilotplan op

*Gebruikte technieken: planningstechnieken, opstellen pilotplan.*Ik kreeg het idee dat deze planning vooral heel belangrijk is als je gekozen hebt voor pilots met een bepaalde functionaliteit als onderwerp. Het is dan heel belangrijk welk onderdeel in welke pilot hoort en wanneer deze wordt uitgevoerd. Aangezien ik een uitgebreide beschrijving van het ontwikkelscenario heb gemaakt, is deze activiteit niet heel belangrijk. De pilots worden in mijn opzet nu niet echt specifieker dan in het ontwikkelscenario staat. Het is in mijn project al duidelijk wat er precies in elke pilots moet komen, voor zover dat duidelijk kan zijn in de stadium.  
Ik zeg overigens niet dat mijn aanpak de beste is en dat dit normaal gesproken geen belangrijk onderdeel is, maar daarover meer in de evaluatie van de strategie.

### 3.1.2 Pilotontwikkeling

#### Pilot 1, deel 1

Plan van aanpak

*Gebruikte technieken: Activiteitenplannning.*Voor het plan van aanpak van dit onderdeel was het vooral belangrijk om vast te leggen welke producten ik wilde hebben voor deze pilot. Ik heb wel vaker een datamodel gemaakt, dus het was niet nieuw voor me. Ik wist al welke onderdelen ik nodig had. Wat ik wel wilde is dat mijn producten gecontroleerd werden door een senior software developer. Hier heb ik een afspraak voor gemaakt en in het plan gezet.

Ontwerp software bouweenheden

*Gebruikte technieken: Entiteit/relatie-modellering, procesbeschrijving.*  
Ik heb in deze activiteit uitgebreid beschreven hoe ik van het systeemconcept naar het datamodel toe heb gewerkt. Het is niet zo dat het altijd het ander oplevert. De een zal kiezen voor de ene oplossing, terwijl een ander niet een andere oplossing zal kiezen. Het is ook niet zo dat de ene oplossing altijd beter is dan de ander. Ik ben tevreden over de techniek en het gebruik van de techniek. De senior die mijn model heeft beoordeeld, was het voor het grootste gedeelte eens met mijn keuzes. Omdat hij meer afwist van het systeem, kon hij toch enkele aanwijzingen geven voor verbetering van het model.

##### Bouw software bouweenheden

Gebruikte technieken: genereren van implementatiemodel.  
Vanuit het UML-programma kon ik het implementatiemodel genereren in de vorm van een script. Dit script kon ik vervolgens uitvoeren op de betreffende servers. Ik heb hierbij geen keuzes hoeven te maken.

#### Pilot 2, deel 1

Plan van aanpak

*Gebruikte technieken: Activiteitenplannning, workshopvoorbereiding*  
Het grootste gedeelte van het plan van aanpak van deze pilot was bestond uit het plannen van de workshop. De opdrachtgever had op eigen initiatief werknemers gevraagd of ze mee wilden werken aan een workshop/hackaton. Ik was natuurlijk blij met de hulp, maar het betekende wel dat ik het een en ander moest regelen. Ik wilde dat iedereen vrijwel direct aan de slag kon. Door het concept op te delen kon ik mensen in teams of individueel hieraan te laten werken.  
Het plan ging bij pilot 2 niet verder dan de hackaton. Dit omdat ik had van tevoren had bepaald dat ik na de hackaton een inventarisatie zou doen en verder zou plannen.

Ontwerp software bouweenheden

Gebruikte technieken: procesbeschrijving  
Ik hoefde voor de front end de bouweenheden, de templates, niet te ontwerpen. Deze worden uit het systeemconcept gehaald. Daarom heb ik deze activiteit gebruikt om uit te leggen hoe dat wordt gedaan. Ik denk dat dat voldoende duidelijk is geworden.

Bouw software bouweenheden

*Gebruikte technieken: workshop, gestructureerd programmeren, interface design, planningstechnieken.*Deze activiteit begon met de hackaton. Deze workshop had als doel een werkend prototype te programmeren. Ik ben erg tevreden over de keuze een workshop te gebruiken als techniek. Dit heeft mij veel programmeerwerk gescheeld. Ik heb nu ook genoeg geprogrammeerd om aan te tonen dat ik deze competentie bezit. Na de workshop was er een werkend prototype en kon ik inventariseren wat er nog gedaan moest worden. Ik heb meerdere keren gedurende het project de MoSCoW techniek gebruikt om te prioritiseren, en heb gemerkt dat dit een erg handige methode is.  
De weken hierna heb ik samen met een collega het prototype verder gebouwd. Ik denk zelf dat de front end goed ontwikkeld. Ik ben geen echte problemen tegengekomen tijdens het bouwen. Het product van deze activiteit is echt geworden als wat ik ervan had verwacht.

#### Pilot 2, deel 1

##### Plan van aanpak

*Gebruikte technieken: inventarisatie benodigde onderdelen.*  
Het bouwen van de back end is voor mij ook een bekend proces. Ik weet welke onderdelen nodig zijn. In dit deel van het plan van aanpak heb ik dan ook geïnventariseerd welke onderdelen nodig zijn in de back end. Ik heb geen problemen ondervonden in deze activiteit.

##### Ontwerp en bouw software-bouweenheden

Gebruikte technieken: Gestructureerd programmeren.  
Ik wil allereerst zeggen dat een groot gedeelte van het programmeren, waar deze activiteit uit bestaat, gedaan is door de deelnemers van de hackaton, mijn tijdelijke A-team. Om aan te tonen dat ik deze competentie wel bezit heb ik uitgebreid beschreven hoe de back end tot stand komt.

##### Test pilotdeel

Gebruikte technieken: regressietests  
 In dit onderdeel heb ik beschreven hoe de regressietests op Hyves worden gemaakt. Ik vind het nooit zo leuk om testen te maken, maar het is wel belangrijk om te doen. Ik ben geen problemen tegengekomen tijdens deze activiteit.

#### Invoering voor testgroep

Het verplaatsen van het product van de development omgeving naar de productieomgeving is niet zoveel moeite. Het invoeren voor de testgroep bestond dan ook vooral uit het op de hoogte stellen van de betrokken personen. Toen dit gedaan was, zijn de pilots ingevoerd voor de testgroep. Hierbij zijn ook geen problemen ontstaan. Dit komt vooral omdat het een proces is dat vaak voorkomt op Hyves. Het werken met capabele collega’s is erg waardevol.

#### Pilot 1,2 en 3: deel 2 (bugfixes)

*Gebruikte technieken: gestructureerd programmeren*Omdat ik gekozen heb voor zeer korte iteraties (behalve de eerste iteratie), is er niet veel ruimte om ongeplande activiteiten tussendoor te doen. Het is zaak om alle aspecten van een project goed in te plannen. Ik ben dan ook erg blij dat ik gekozen heb om één iteratie te gebruiken voor bugfixes. Had ik dat niet gedaan, dan was ik in tijdnood gekomen. Ik zou als ik weer een planning zou moeten maken ook weer een week inplannen voor het fixen van bugs. Mocht je geen week nodig hebben, kan dan gewoon gestart worden met de volgende iteratie.

#### Pilot 2 en 3: deel 3, 4, 5.

*Gebruikte technieken: gestructureerd programmeren, interaction design, activiteitenplanning.*Ik heb de laatste iteraties van pilots 2 en 3 in de evaluatie samengevoegd. Dit heb ik gedaan omdat ze qua opzet en functie hetzelfde zijn. Van tevoren was er voor deze pilots nog geen onderwerp bepaald. Ik vond het heel prettig werken om zelf een systeemconcept te bepalen, in te voeren en daarna aan te passen met de feedback van testgebruikers. Ik denk ook dat het een ideale combinatie is van het gebruiken van de expertise van ontwikkelaars om een systeem neer te zetten en het bijschaven van dat systeem aan de hand van de wensen van gebruikers. Uiteraard moet de opdrachtgever wel de mogelijkheid toe geven. Ik denk dat niet veel opdrachtgevers erg blij zullen zijn als je zegt dat je niet precies weet wat je de maand na de eerste invoering gaat doen. Ik had bij dit project wel die mogelijkheid, en heb daar dankbaar gebruik van gemaakt.

### 3.1.4 Invoering

Bij de invoering is het het belangrijkste dat iedereen die iets met het product te maken heeft, weet wanneer de invoering precies is. Mocht er dan wat misgaan, kun je zo snel mogelijk handelen. Aangezien er al een soft launch was geweest voor Hyves Voorspellen, was de kans op problemen al flink afgenomen. Er zijn bij de invoering voor het grote publiek dan ook geen problemen geweest.

##### Evalueer testresultaten

Deze activiteit is gepland na de overdracht. Het uitwerken en conclusies verbinden aan de testresultaten is iets wat ik niet zelf doe. Ik beperk me in deze activiteit alleen tot het kijken naar de resultaten. Als ik dan zie dat er gemiddeld 150.000 deelnemers zijn is dat een mooi resultaat. Daarbij komen nog de honderden of soms duizenden reacties. Er zijn wedstrijden geweest met tienduizenden reacties. Ik ben dan ook erg tevreden met de resultaten.

### 3.1.5 Strategie

Ik heb dubbele gevoelens over de pilotiteratiestrategie. Nadat ik voor de IAD methode heb gekozen heb ik enkele voorbeelden opgezocht van scripties die deze methode gebruikten. Hier werden de 3 tiers uit het 3-tier-model gebruikt als pilots. Ik dacht dat dat de bedoeling was en heb dit zo opgenomen in mijn afstudeerplan. Later, toen ik al een tijdje bezig was, ben ik er achter gekomen dat het eigenlijk beter om pilots te definiëren als ‘taartpunt’ uit het 3-tier model. Mijn iteraties na de soft launch zijn eigenlijk ook in deze vorm uitgevoerd. Ik denk met dat de taartpunt-strategie de hoeveelheid werk beter te verdelen is over de pilots. In mijn project was de ene pilot aanzienlijk groter dan de andere. Het is nu goed gegaan, zonder veel problemen, maar ik zou toch een volgende keer voor de taartpunt-strategie kiezen.  
Ik ben wel erg te spreken over de iteraties na de soft launch. Omdat deze nog niet vanaf het begin een onderwerp toegewezen hebben gekregen, kon ik deze invullen met systeemeisen die pas later in het project waren geformuleerd.

### 3.1.6 Methode

Het IAD model heb ik als een prettige methode ervaren. Omdat het de eerste keer was dat ik deze methode gebruikte heeft het even geduurd om door te hebben wat exact de belangrijke onderdelen zijn. Ik bedoel daarmee, dat ik een dik boek vol met aanwijzingen en IAD regels heb. Als ik alles precies doe zoals in het boek beschreven staat, is het te verzetten werk buiten proporties met de schaal van het project. Als je de methode begrijpt geeft het een erg handige leidraad voor een project.  
Ik vond het ook aardig om te zien dat deze methode en strategie prima in te passen zijn in de ontwikkelmethode die op Hyves gebruikt worden. De hybride vorm die ik gebruikt heb, met systeemeisen die steeds duidelijker worden, leek sterk op hoe er bij Hyves wordt ontwikkeld. IAD schrijft iets langere iteratiecycli voor dan één week. Hyves is echter zo’n veranderlijk product, dat het echt nodig is om elke week te releasen. Met deze aanpassing paste de IAD methode prima in de ontwikkelomgeving van Hyves.

## 3.2 Productevaluatie

### 3.2.1 opdracht

Het resultaat van de opdracht is enerzijds het werkende Hyves Voorspellen als product, en anderzijds het adviesrapport. Deze twee producten zijn duidelijke eindresultaten die behaald zijn. Ik heb echter ook geschreven dat ik wilde kijken of er groei in de diepte mogelijk was. Ik wilde kijken of mensen door Hyves Voorspellen meer tijd doorbrengen op Hyves. Tijdens het project ben ik er wel achter gekomen dat dit niet zo makkelijk te meten is. Er zijn zoveel andere onderdelen op Hyves die het gebruik kunnen beïnvloeden en tegelijkertijd worden ontwikkeld als mijn Hyves Voorspellen. Hiernaast spelen acties als activiteiten van concurrenten als Facebook een rol. Zelf een artikel in de pers kan het gebruik erg beïnvloeden. Ik zal in de toekomst dan ook iets voorzichtiger zijn bij het schrijven van de opdrachtsomschrijving.

### 3.2.2 Doelstelling

De doelstelling was:

**Adviseren over de vraag of het mogelijke is communicatie te sturen naar events in de toekomst, met het ontwikkelen van het Hyves Voorspellen platform als casus.**

Ik denk dat dit een goede doelstelling is, en dat de doelstelling is behaald. De eerste meetresultaten zijn al een eerste indicatie om deze vraag te beantwoorden. Als er niet veel voorspellingen gedaan zouden zijn, zou dat ook een antwoord op de vraag kunnen zijn. In dat geval zijn er nog andere zaken die uitgesloten moeten worden, zoals: ‘is het systeem goed?’ of ‘moet er reclame voor worden gemaakt?’. Ik denk dat ik door dit project heb kunnen laten zien dat het systeem goed is gemaakt.

### 3.2.3 Probleemstelling

De probleemstelling was:

**Hyves is op zoek naar groei, maar kan dit niet doen door nieuwe leden aan te trekken.**

Deze probleemstelling is denk ik correct bij dit project. Ik denk wel dat deze doelstelling voorlopig de focus van Hyves zal hebben. De concurrentie is groter dan ooit, en Hyves moet zijn best doen om zijn leden te behouden. Dat betekent dat Hyves Voorspellen als een onderdeel moet worden gezien van een richting die uitgestippeld is door de directie van Hyves. Doe dingen die de concurrent niet doet of kan doen. Een van die dingen is een voor Nederland op maat gemaakt voetbalvoorspellingensysteem.

Naar aanleiding van de goede resultaten van Hyves Voorspellen, is Hyves inmiddels een ander project gestart: ‘Hyves Nu & Straks’. Het idee daarvan is gebaseerd op het praten over actuele onderwerpen of onderwerpen in de toekomst, zoals tv-programma’s. Een ander format dat zal worden gemaakt naar aanleiding van Hyves Voorspellen is het voorspellen van de ontwikkelingen in het programma “Wie is de mol?”.

# Literatuurlijst:

Boek:

Het evolutionair ontwikkelen van informatiesystemen. R.J.H. Tolido.

Internet:

<http://developers.facebook.com/blog/post/358/>

<http://en.wikipedia.org/wiki/Release_early,_release_often>

<http://nl.wikipedia.org/wiki/Entity-relationshipmodel>

<http://faculty.washington.edu/jtenenbg/courses/360/f04/sessions/schneidermanGoldenRules.html>

<http://www.useit.com/papers/heuristic/heuristic_list.html>

<http://nl.wikipedia.org/wiki/Abstractielaag>

<http://gamification.co/2011/03/15/the-social-psychology-of-badges/>

<http://fbindie.posterous.com/daumau-crash-course-the-main-measure-of-game>

[http://techcrunch.com/2009/10/29/how-to-measure-the-true-stickiness-and-success-of-a- facebook-app/](http://techcrunch.com/2009/10/29/how-to-measure-the-true-stickiness-and-success-of-a-%20%20facebook-app/)

<http://www.insidesocialgames.com/2009/10/27/the-sticky-factor-creating-a-benchmark-for-social-gaming-success/>

# Verklarende woordenlijst

Badge Beloning voor een prestatie. Bestaat uit een plaatje met beschrijving op het digitale profiel van de gebruiker.

Buzz item Een bericht uit de lijst met updates op de homepage van Hyves.

Class Het skelet van een object in een programmeertaal.

Deployen Zie ‘Live zetten’.

Datamodel Een model dat een weergave is van een bepaalde database.

GANTT chart Strokenplanning.

Hackaton Workshop waarbij aan een product gewerkt wordt. Semi-informeel.

Heuristic Op ervaring gebaseerde richtlijn.

HR Human Resources.

Live zetten Aanpassingen van – of een heel systeem verplaatsen naar de productie omgeving.

Memcached Bepaalde cachingmethode.

Mockup Beeldschermontwerp van een bepaalde webpagina.

MySQL Databasesysteem. SQL is de programmeertaal hiervoor.

PHP Programmeertaal.

Propel Programma dat gebruikt wordt om databases te beheren etc.

Property-bag Een kolom in een database waarin meerdere waarden op zijn geslagen.

Python Programmeertaal.

QA Quality Assurance.

Requirements Eigenschappen van een systeem die vereist zijn.

Regressietest Test om te controleren of bestaande functionaliteit blijft werken na aanpassing.

RFC Request For Comments. Verzoek of mensen hun mening willen geven.

Soft launch Het toegankelijk maken van een bepaald product voor een beperkte groep gebruikers.

Smarty Programmeertaal om markup en Business logic te scheiden.

UML Unified Modeling Language. Taal om diagrammen weer te geven.

Wireframes Zeer schematische weergave van een beeldschermontwerp.