

Requirements Engineering: League of Legends eSports

Bellemans Jonah
Lambrichts Midas

Caerts Stijn
Verhoest Elise

Dimova Yana
Vos Pieter

22 december 2016

Inhoudsopgave

1	Motivatatie sportkeuze	2
2	Beschrijving Competitie	2
2.1	Plaatsingsmatches	2
2.2	Wedstrijdverloop	2
2.3	Competitieverloop	2
2.3.1	Eerste competitieronde: Group Stage	2
2.3.2	Tweede competitieronde: Quarterfinals	3
2.3.3	Derde competitieronde: Semifinals	3
2.3.4	Finale	3
3	Systeemcontext	4
4	Doelen	7
4.1	Dependencies	7
4.2	Rationale	9
5	Scenarios	9
5.1	“Planning opstellen”	9
5.2	“Planning updaten”	10
5.3	“Planning wijzigen”	10
5.4	“Ploeg aanmaken”	11
6	Datamodel	11
6.1	UML diagram	11
6.2	OCL Constraints	12

1 Motivatie sportkeuze

We opteerden om e-sports, meer bepaald de “League of Legends World Championships (LoL Worlds)”, te kiezen als sporttak voor dit project. Wij kozen hiervoor omdat deze sporttak origineel is, en een zeer duidelijke afbakening van de spelen wedstrijdreglementen voorziet. Aangezien de meerderheid van de groepsleden vanuit een computer science-gerelateerde achtergrond komt, is de e-sport goed door de leden gekend, en is iedereen gemotiveerd om hieraan te werken.

2 Beschrijving Competitie

2.1 Plaatsingsmatches

Teams plaatsen zich voor de competitie door middel van plaatsingswedstrijden die vroeger in het seizoen plaatsvinden. Voor elke regio (West-Europa, Noord-Amerika, Korea, . . .) worden een aantal “Seeds” vrijgehouden, die worden ingevuld door de beste ploegen van die regionale competities. De absolute topploegen van de wereld, zoals winnaars van eerdere jaren, kunnen via een “wildcard” rechtstreeks uitgenodigd worden voor deelname door het organisatorisch comité.

2.2 Wedstrijdverloop

Elke wedstrijd van het spel League of Legends (“LoL”) wordt gespeeld door twee teams van elk vijf spelers. De teamleden besturen elk een individueel personage in de game, die men “Champions” noemt. Na het selecteren van de champions voor elke speler, worden de teams in de spelwereld (“Summoner’s Rift”) gezet. Het doel van het spel is om de “Nexus” van het andere team te vernietigen, terwijl men de eigen Nexus verdedigt. Het spel blijft lopen totdat er de Nexus van één van beide teams vernietigd is. Het is dus onmogelijk om een match te beëindigen zonder dat er een éénduidige winnaar bepaald is.

2.3 Competitieverloop

2.3.1 Eerste competitieronde: Group Stage

De wereldkampioenschappen beginnen met 16 verschillende teams, die opgedeeld worden in groepen. Binnen de groep spelen de teams telkens één tegen één. De “matchups” worden bepaald met een round-robin systeem, waarbij elk team tweemaal niet-openvolgend tegen ieder ander team speelt (eenmaal aan elke zijde van Summoner’s Rift).

Tiebreaker Als twee teams dezelfde score behalen, meer bepaald, hetzelfde win-percentages behalen, zal de zogenaamde “head-to-head” score (de score die door de teams behaald werd tijdens hun onderlinge wedstrijden) gebruikt worden om een winnaar te bepalen. Als ook hieruit geen winnaar besloten kan worden, zal een enkele tiebreak match gespeeld worden.

2.3.2 Tweede competitieronde: Quarterfinals

Tijdens de kwartfinales (“Quarterfinals”) blijven slechts acht seeds over, die ingenomen worden door de twee hoogst scorende teams uit iedere groep van de Group Stage. De teams die #1 werden binnen hun groep, spelen tegen de teams die #2 werden in de andere groep. Welke groepen dan juist gecombineerd worden, wordt willekeurig bepaald. Eens er een matchup bepaald is, ligt daarmee ook de “samenhangende” matchup vast. Als de willekeurige trekking dus bepaalt dat Team 1 uit groep A speelt tegen Team 2 uit groep D, dan zal ook Team 2 uit groep A spelen tegen Team 1 uit groep D. Nadat alle matchups vastliggen, spelen de gekozen teams telkens een “best-of-five” match, waaruit de winnaar doorstroomt naar de volgende ronde.

2.3.3 Derde competitieronde: Semifinals

In de halve finale (“Semifinals”) spelen de winnende teams uit de kwartfinales opnieuw een “best-of-five” match tegen elkaar. De twee overblijvende winnende teams gaan vervolgens naar de finale.

2.3.4 Finale

De finale wordt gespeeld door de twee overgebleven teams. Ook hier wordt de uiteindelijke winnaar bepaald door een “best-of-five” match. Het team dat op het einde van deze vijf games het meeste gewonnen heeft, is de uiteindelijke winnaar van het toernooi.

3 **Systeemcontext**

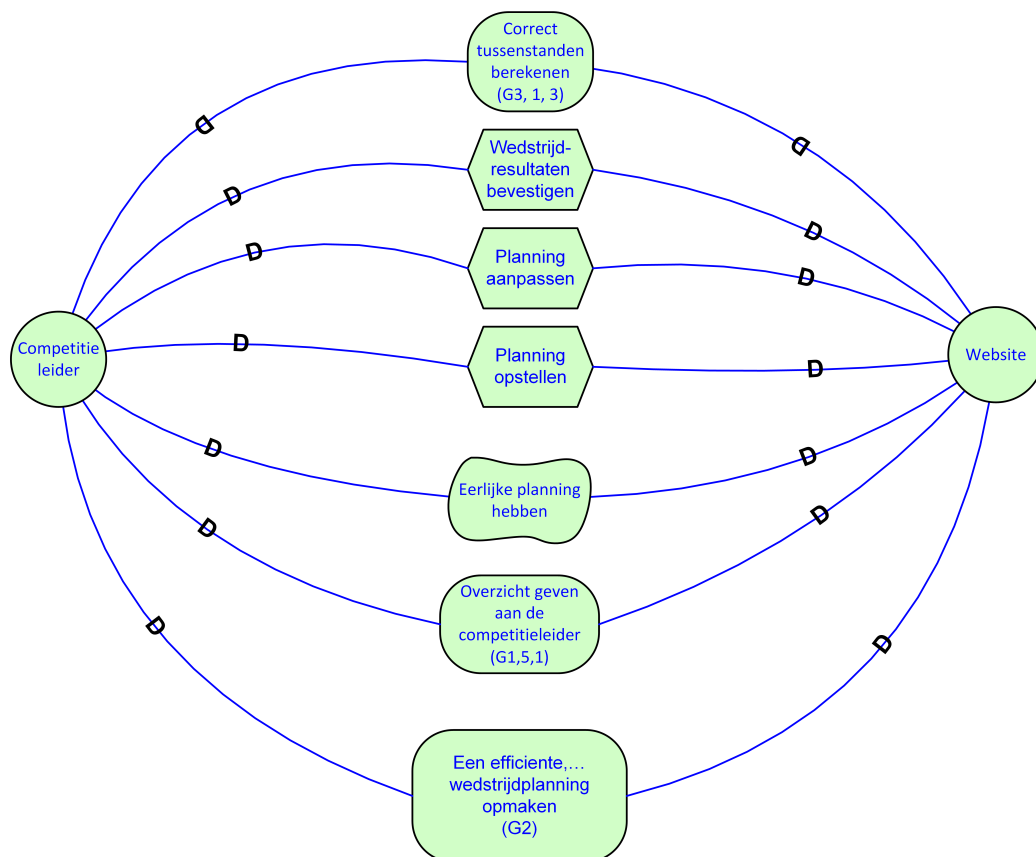
- Subject Context Facet
 - Requirement Sources
 - * Stakeholders
 - Wedstrijdleider
 - Competitieleider
 - Spelers
 - Scheidsrechters
 - Shoutcasters
 - * Documentatie
 - World Championship rule set
 - Leaguepedia
 - League of Legends Terms of Use
 - * Bestaande systemen
 - Context Objects
 - * Spelers
 - * Scheidsrechters
 - * Stadium
 - * Materiaal (pc's, toetsenborden, muizen)
 - * Internetconnectie
 - * Internetsnelheid
 - * Publiek
 - * League accounts
 - * Security
 - Properties and relationships
 - * Efficiente planning van de wedstrijden
 - * Correctheid van de planning
 - * Respect tegenover het reglement
 - * Beschikbaarheid van de spelers
 - * Beschikbaarheid van de zalen
 - * Beschikbaarheid van de scheidsrechters
 - * Beschikbaarheid van de shoutcasters
 - * Werking van het materiaal

- * Werking van het internet
 - * Veiligheidsmaatregelen
- Usage Context Facet
 - Requirement Sources
 - * Stakeholders
 - Spelers
 - Publiek
 - Scheidsrechters
 - Competitieleider
 - Wedstrijdleider
 - Shoutcasters
 - * Documentatie
 - * Systemen
 - Andere systemen die resultaten van een competitie bijhouden (voor een andere sport bijv.)
 - Context Objects
 - * Publiek
 - * Spelers
 - * Competitieleider
 - * Wedstrijdleider
 - * Scheidsrechters
 - * Shoutcasters
 - Properties and relationships
 - * Gebruik van de user interface
 - * ...
- IT Context Facet
 - Requirement Sources
 - * Stakeholders
 - * Documentatie
 - * Systemen
 - Context Objects
 - * Website

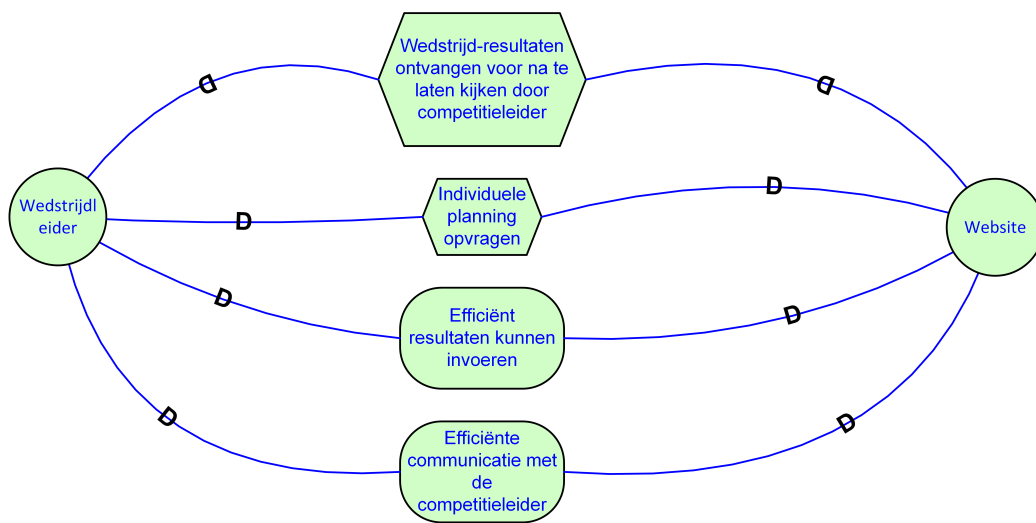
- * Internet
 - * User interface
 - * User database
 - * Scheidsrechters database
 - * Zalen database
 - * Resultaten database
 - * Streammogelijkheden
- Properties and relationships
 - * Updaten van resultaten
 - * Correct bijhouden van gegevens in databases
 - * Live uitzending van de stream
 - * Beschikbaarheid van website
 - * Gebruiksvriendelijkheid van user interface
- Development Context Facet
 - Requirement Sources
 - * Stakeholders
 - Context Objects
 - Properties and relationships

4 Doelen

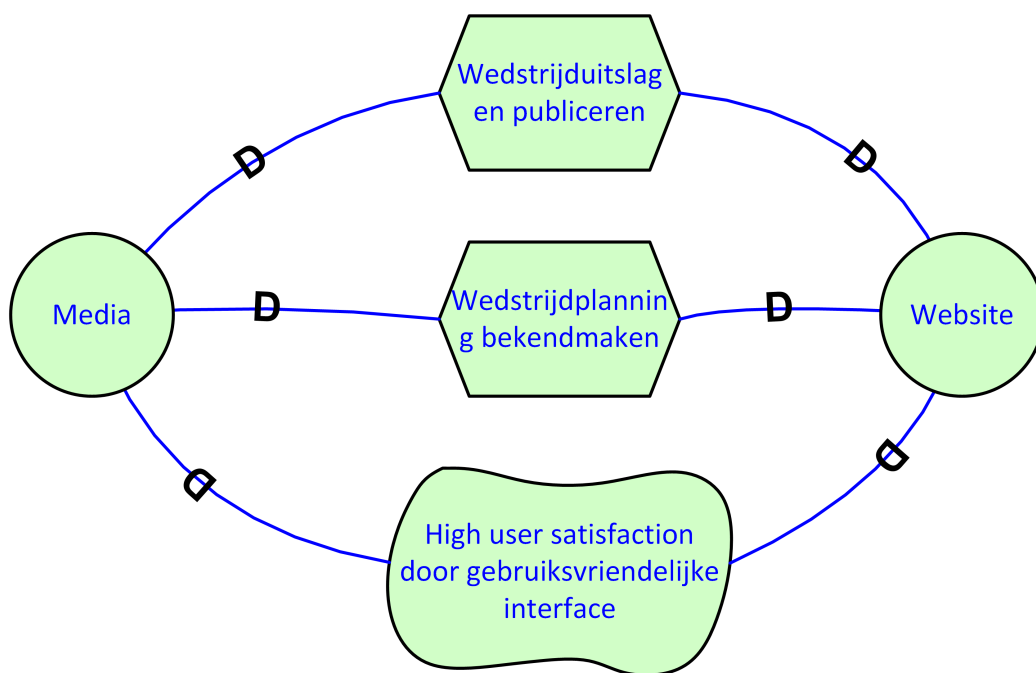
4.1 Dependencies



Figuur 1: Eerste Dependency model in i*

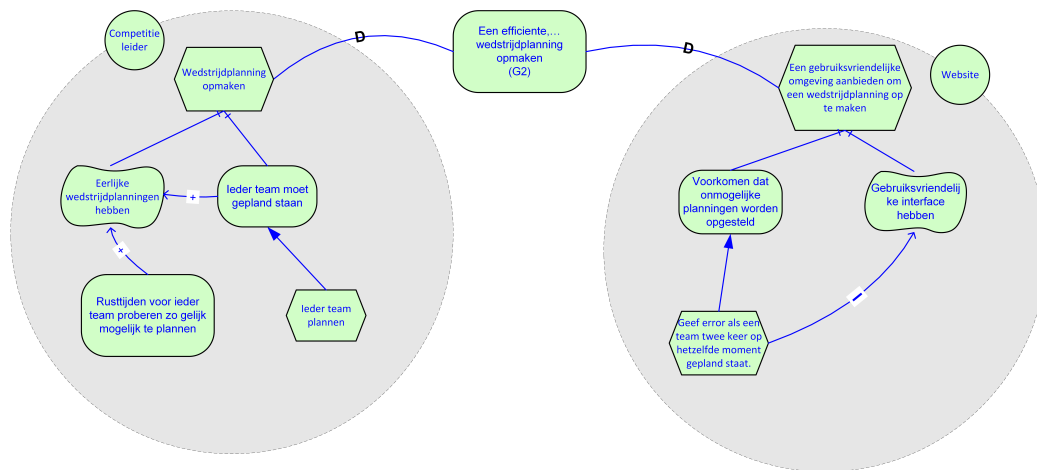


Figuur 2: Tweede Dependency model in i*



Figuur 3: Derde Dependency model in i*

4.2 Rationale



Figuur 4: Strategic Rationale model in i*

5 Scenarios

5.1 “Planning opstellen”

De competitieleider wil een nieuwe match tussen 2 ploegen in de planning zetten. Hij gaat naar de website, drukt op “log in” en voert zijn persoonlijke inloggegevens in. De competitieleider heeft de optie om de wedstrijdplanning aan te passen of een nieuwe match toe te voegen op de website. Hij gaat naar de webpagina waar hij een nieuwe wedstrijd kan toevoegen via de knop “wedstrijd toevoegen”. Vervolgens krijgt hij een overzicht van de beschikbaarheden van de ploegen, plaatsen, tijdstippen, shoutcasters, scheidsrechters. (Hij ziet een paar velden waar hij de naam van de ploegen, de plaats, het tijdstip, de scheidsrechter en de shoutcasters nodig voor de match kan invullen.) Hij vult de namen van de 2 ploegen in die hij tegen elkaar wilt laten spelen. Hij selecteert een plaats tussen de mogelijkheden opgeslagen in de database. De competitieleider voegt vervolgens een beschikbaar tijdstip toe, alsook een scheidsrechter uit de database die op het gegeven tijdstip beschikbaar is. De competitieleider wijst ook shouters toe voor de wedstrijd. De competitieleider slaat de nieuwe gegevens op en sluit de website.

5.2 “Planning updaten”

1. De competitieleider opent de website
2. De competitieleider drukt op “log in”
3. De competitieleider meldt zich aan met zijn persoonlijke inloggegevens
4. De competitieleider klikt op de knop “wedstrijdplanning aanpassen”
5. De competitieleider klikt op de knop “alle geplande wedstrijden weergeven”
6. De competitieleider klikt op de planning die hij wilt aanpassen
7. De competitieleider klikt op de knop “invoeren ploegnaam”
8. De competitieleider voert de naam van de ploeg in, in het veld “Ploegnaam”
9. De competitieleider klikt op de knop “wijzigingen opslaan”
10. De competitieleider klikt op de knop “afmelden”
11. De competitieleider sluit de competitiewebsite

5.3 “Planning wijzigen”

De competitieleider probeert op de planning op de website een timeslot te wijzigen dat al voorbij is:

1. De competitieleider opent de website
2. De competitieleider drukt op “log in”
3. De competitieleider meldt zich aan met zijn persoonlijke inloggegevens
4. De competitieleider klikt op de knop “wedstrijdplanning aanpassen”
5. De competitieleider klikt op de knop “alle geplande wedstrijden weergeven”
6. De competitieleider klikt op de planning die hij wilt aanpassen
7. De competitieleider klikt op de knop “wijzig tijdstip”
8. De competitieleider klikt op een timeslot dat reeds voorbij is.
9. Er opent een waarschuwingsvenster waar opstaat: ‘dit timeslot kan niet aangepast worden: datum is al voorbij’.
10. De planning blijft onveranderd.

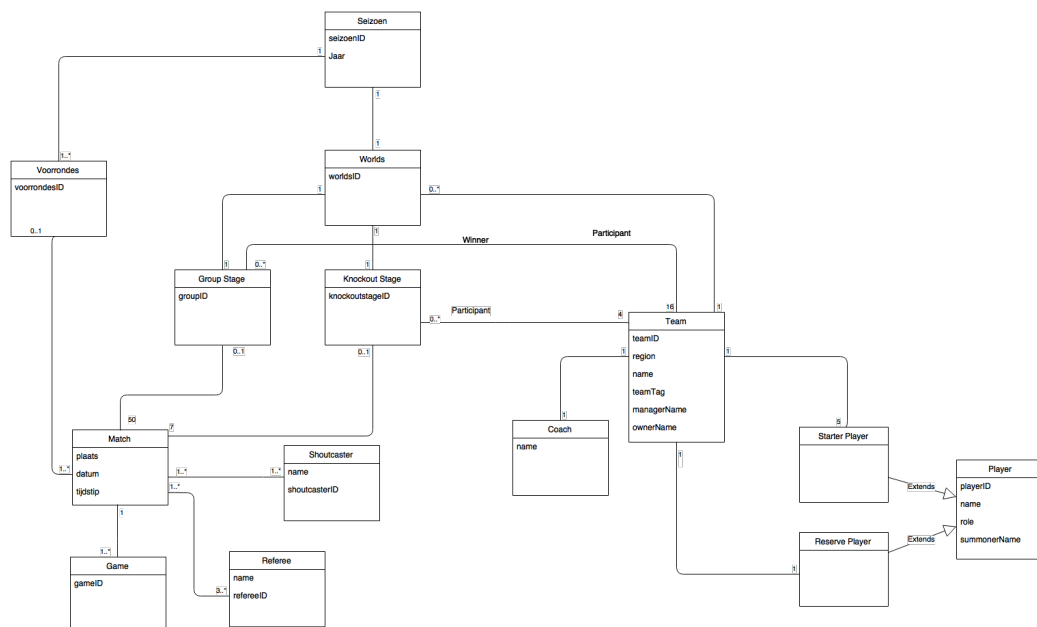
5.4 “Ploeg aanmaken”

Parth Naidu, de coach van ploeg Team SoloMid, wilt Sneaky, een speler van ploeg Cloud 9, toevoegen aan zijn team:

Parth Naidu, de coach van ploeg Team SoloMid, opent de website. Hij klikt op “log in” en logt vervolgens in met zijn persoonlijke inloggegevens. Vervolgens ziet hij een link naar “ploegen beheren”, waar hij dan ook op klikt. De site brengt hem dan naar een pagina waar hij zijn ploeg kan beheren, op die pagina staat ook een link om een speler toe te voegen. Hij klikt op deze link en krijgt vervolgens een formulier dat hij kan invullen om een nieuwe speler aan zijn ploeg toe te voegen. Hij vult de gegevens van Sneaky, een speler van ploeg Cloud 9, in op het formulier. Hij probeert dit ingevulde formulier in te vullen. De website weigert deze verandering door te voeren en laat een waarschuwing zien aan Parth Naidu, waar op staat: “Deze speler speelt al voor een andere ploeg, hij mag niet voor 2 ploegen tegelijk spelen. Hij moet eerst de andere ploeg verlaten vooraleer hij bij een andere ploeg kan toegevoegd worden.”

6 Datamodel

6.1 UML diagram



Figuur 5: UML representatie van het model

6.2 OCL Constraints

```
context Worlds invariant  
group_stage.Participant ->  
    includesAll(knockout_stage.participant)
```

Alle teams van de knockout stage moeten ook aan de group stage hebben meegedaan.

```
context Worlds invariant  
Knockout_stage.Participant -> includesAll(winner)
```

De winnaar van worlds moet hebben meegedaan in de knockout stage.

```
context Team invariant  
starter_player -> excludeAll(reserve_player)
```

Een speler is oftewel een starter player oftewel een reserve player, hij kan niet beide tegelijk zijn.