

# Plan van aanpak voor xMAS ontwerptool

Guus Bonnema, Stefan Versluys, Jeroen Kleijn

21/10/2014

Wijziging 10 februari 2015

## Samenvatting

Dit document is het plan van aanpak voor het ABI (Afstudeerproject bachelor informatica). De opdrachtgever is Bernard van Gastel en de begeleider is Freek Verbeek. Het project is gericht op het verbeteren van een bestaand chip ontwerptool op de aspecten platformafhankelijkheid, uitbreidbaarheid, en integratie met de verificatietools.

Het plan is gebouwd op basis van een hybride agile aanpak, een aanpak die moet passen bij de noden van het team alsook voor het probleem.

De domeinanalyses omvatten onderzoek van deelgebieden uit de productcontext dat alternatieven voor oplossingen aan het licht moeten brengen. Op deze manier onderbouwen wij onze keuze op een wetenschappelijke manier en kunnen we verantwoord advies geven aan de opdrachtgever.

De onderzoekscontext vindt plaats tijdens een later deel van het project zodat het team de tijd heeft om materie beter te leren kennen en het verband tussen project en onderzoek beter kan verwoorden.

Het resultaat van het project is een gedocumenteerd ontwerptool, dat voor 3 platformen gebouwd is en goed integreert met de bestaande tools voor verificatie.

### Aanpassing febr 2015

Naar aanleiding van nieuwe omstandigheden binnen de OU en de daarmee gepaard gaande vermindering van communicatie sinds december 2014, hebben we het plan aangepast aan de nieuwe omstandigheden en de huidige inzichten. Met name het de sectie over risico's is aangepast. Wijzigingen staan in groene kaders.

De planning blijft ongewijzigd.

## 1.1 Business case

De projectaanvraag voor dit project van Bernard van Gastel en Freek Verbeek verwoordt de business case in de projectaanvraag als volgt.

“De Network-on-Chip (NoC) groep van de OU doet onderzoek naar nieuwe methoden om NoC ontwerpen (moderne manier van ontwerpen van processoren) foutvrij te krijgen met behoud van efficiëntie. De NoC groep heeft een aantal tools ter ondersteuning van het onderzoek gemaakt (met ondersteuning van vele studenten projecten), voornamelijk WickedXmas. Deze tool stelt de gebruiker in staat om chip ontwerp te maken/bewerken/genereren in de visuele xMAS taal van Intel, en vervolgens een aantal door ons ontwikkelde tools op los te laten (bv symbolische analyse, deadlock checker, etc). De huidige tool heeft een aantal problemen:

- niet modulair opgezet (waardoor uitbreidingen moeizaam gaan)
- op Windows API gebaseerd (waardoor de onderzoekers die gebruik maken van Mac het lastig kunnen gebruiken)
- moeizame integratie met tools: WickedXmas is nu geschreven in C#, en lijkt moeilijk te integreren met onze C/C++ tools
- geen documentatie

Deze problemen moeten opgelost worden, danwel door een grote refactoring van de bestaande code, danwel door het opnieuw bouwen.”

Omdat dit het afstudeerproject is van drie studenten, zijn er geen financiële overwegingen. De kosten die er zijn (inzet opdrachtgever en begeleider) vallen onder het afstuderen. Deze kosten zijn al gedekt als onderdeel van de cursus Afstudeerproject Bachelor Informatica (ABI).

De baten zijn vooral betere onderhoudbaarheid en uitbreidbaarheid en een bredere ondersteuning van platformen. Deze verbeteringen ondersteunen met name het chip-ontwerponderzoek aan universiteiten en bedrijven. Dankzij een betere integratie tussen ontwerptool en verificatietools optimaliseren de onderzoekers hun workflow. Het aantal handelingen dat zij moeten uitvoeren is minder dan met het huidige tool en de directe feedback in de user interface versnelt het lokaliseren en verhelpen van fouten in een ontwerp.

## 1.2 Vision

Deze sectie geeft aan wat de opdrachtgever ziet als ideale uitkomst van het project. Het ontwerptool is bedoeld voor wetenschappers van universiteiten en van bedrijven (o.a. Intel). Onderzoekers op verschillende platformen kunnen het ontwerptool gemakkelijk downloaden, installeren en gebruiken. Het tool toont op verzoek tijdens het ontwerp fouten die door de verificatietools zijn gevonden.

Stakeholder	Projectrollen	omgevingsrollen
Bernard van Gastel	Opdrachtgever	Onderzoeker, gebruiker, ontwikkelaar
Freek Verbeek	Begeleider	Onderzoeker, gebruiker, ontwikkelaar
Team33	Ontwikkelaar, student	
Univ. medewerkers		Onderzoeker, onderzoeker NoC, gebruiker
Bedrijfsmedewerkers		Onderzoeker, gebruiker

Figuur 1: Stakeholders en hun rollen

Het programma is goed gestructureerd, uit te breiden met nieuwe verificatietools en NoC onderzoekers kunnen gemakkelijk nieuwe primitieven definiëren. Ten slotte kunnen de xMAS -onderzoekers onze bestanden converteren naar een ander formaat zoals Verilog.

### 1.3 Stakeholders

Opdrachtgever is Bernard van Gastel van de Open Universiteit. Begeleider is Freek Verbeek van de Open Universiteit. De doelgroep voor gebruikers van het ontwerptool bestaat uit medewerkers van universiteiten en onderzoekers van bedrijven zoals Intel en LLC. Een kleine groep van xMAS -onderzoekers van zowel de universiteit als bedrijven gebruiken het ontwerptool vanuit de optiek van onderzoek naar verbetering van NoC ontwerp. Doelstellingen voor deze groep zijn chip import en export van bestandsformaten (zoals Verilog) en verificatie van aspecten van correctheid (deadlock, livelock, syntactische of semantische checks). Bernard en Freek nemen het onderhoud van het tool voor hun rekening.

Stakeholder	Verband	Rol	Freq	Belang	Invloed
Bernard van Gastel	Onderzoek NoC	Opdrachtgever	Hoog	Hoog	Hoog
	Onderhoud	Opdrachtgever	Laag	Hoog	
	Gebruik	Gebruiker		Hoog	
Freek Verbeek	Onderzoek NoC	Begeleider	Hoog	Hoog	Hoog
	Onderhoud	Opdrachtgever	Laag	Hoog	
	Onderhoud	Ontwikkelaar	Laag	Hoog	
	Gebruik	Gebruiker		Hoog	
Univ. medewerkers	Onderzoek alg.	Gebruiker	Laag	Middel	
	Onderzoek NoC	Gebruiker	Middel	Hoog	
	Gebruik			Middel	
Bedrijfsonderzoekers	Onderzoek NoC	-	Middel	Hoog	
	Gebruik	Gebruiker		Middel	
	Installatie	Gebruiker		Hoog	
Bedrijfsmedewerkers	Gebruik	Gebruiker	Laag	Middel	
	Installatie	Gebruiker		Hoog	
Team33	Ontwikkeling	Ontwikkelaar	Hoog	Hoog	Hoog
	Afstuderen	Student		Hoog	Hoog
	Samenwerking	Ontwikkelaar	Hoog	Hoog	Hoog

## 1.4 Challenges and goals

Hieronder een lijst van uitdagingen (challenges). Tijdens de voorbereidingsfasen werken we de uitdagingen uit tot high level requirements.

**platformen** Het onderzoeksteam werkt met Mac OS, ontwikkelteam met MS Windows en Linux, de opdrachtgever en studenten met MS Windows, Linux of Mac OS. Dit genereert requirements voor platformonafhankelijkheid van ontwikkelomgeving, taal en toolkits die we gebruiken.

**integratie** Het systeem bestaat uit meerdere modules, grofweg aangeduid als ontwerp-tool (wat ons onderwerp is) en verificatietools (tools die deel gebouwd en deels in ontwikkeling zijn). De uitdaging is een hoge graad van integratie van deze componenten, zo onafhankelijk mogelijk. Dit genereert requirements voor onderhoudbaarheid, componentinterfaces en applicatiestructuur. De verificatietools zijn alle C++ programma's. De interfaces zijn gebaseerd op JSON.

**onderhoud** Over de tijd heen onderhouden vele mensen de software, die elkaar niet spreken. Dit genereert requirements met betrekking tot documentatie, onderhoudbaarheid en toegankelijkheid.

**Uitbreiding** De huidige software is moeilijk uitbreidbaar op de wijze die de opdrachtgever graag wil. De wens is relatief gemakkelijk een nieuwe verificatiemodule te kunnen toevoegen.

**functie** Het ontwerptool ondersteunt de primitieven en hun samenwerking zoals gespecificeerd in het xMAS paper [CKO12]. Het huidige ontwerp tool: WickedXMas geeft de richting aan implementatie van de functionaliteit en de interface, maar is niet beperkend of maatgevend daarvoor.

#### Aanpassing febr 2015

**Verminderde communicatie** Sinds december 2014 is de communicatie drastisch verminderd. Eind januari 2015 licht Bernard toe waarom: organisatie wijzigingen. Het gevolg is dat Bernard 50% van zijn tijd les gaat geven en de lessen op korte termijn van vervroegd gepensioneerde docenten overneemt. Freek gaat 3 maanden naar Amerika. Dit heeft gevolgen voor de intensiviteit van communicatie. Met name in een agile omgeving kan dit consequenties hebben. Eén gevolg hebben we al gemerkt: in december moesten we aannames doen over het datamodel, dat we eind januari weer konden terugdraaien na overleg met Bernard.

## 1.5 Succesfactoren

Het project is een succes als

- het ontwerptool een ontwerp kan maken in de xMAS taal en de verificatietools kan aanroepen voor controle.
- Gebruikers van het tool op de volgende platforms kan draaien:
  - MS Windows vanaf versie 7 64 bits (32 bits via emulatie)
  - Linux 64 bits en 32 bits
  - Mac OS X (versie?)

Het project is een groot succes als het project een succes is en:

- De gebruiker kan validatie- en verificatiesoftware aanzetten of uitzetten
- De gebruiker kan nieuwe primitieven maken en aankoppelen.

Het project is mislukt als het ontwerp tool geen succes is.

## 1.6 Risico

De onderkende risico's zijn vooral projectrisico's. De meeste projectrisico's komen voort uit de geografische spreiding van het team (i.e. de studenten) en opdrachtgever en begeleider. Diverse maatregelen zijn dan ook gericht op het verminderen van de risico's die de spreiding veroorzaakt.

### Aanpassing febr 2015

Sinds december 2014 komen daar de risico's als gevolg van communicatie vermindering bij: zie sectie 1.4. Eén van de gevolgen kan zijn onvoldoende of trage feedback met als gevolg een eindproduct dat minder goed op de wensen is afgestemd binnen de gestelde planning. Een ander gevolg kan zijn dat we deels werk opnieuw moeten doen omdat we uitgaan van aannames.

Het gaat om herstructureren dan wel herbouwen van bestaande programmatuur. Daarom zijn er weinig systeemrisico's. De inhoudelijke kennis is bij de opdrachtgever en de begeleider aanwezig. Het grootste systeemrisico is het kiezen van een user interface toolkit en de ontwikkelomgeving. Dit onderdeel bepaalt de platformafhankelijkheid en heeft invloed op de architectuur. Het plan besteedt apart aandacht aan het onderdeel in een domeinanalyse om vroegtijdig dit risico weg te nemen.

Zie figuur 1.6 voor een overzicht van de risico's die we onderkennen en figuur 1.6 voor een overzicht van de maatregelen die we nemen ter compensatie van risico's. Voor elk risico op één na hebben we een maatregel bedacht. Het risico voor fouten in het ontwerp accepteren we omdat deze ontwerpen globaal en richtinggevend zijn. Voor de productie van chip ontwerpen begint het werkelijke ontwerpproces in groter detail dan ons tool biedt.

Tabel 1: Relevante risico's voor dit project

risico		
	oorzaak	maatregel
1	<i>vertraging in het project of de oplevering</i>	
	Stakeholders niet beschikbaar wanneer dat gewenst is (gerealiseerd risico) .	skype(1),email(2),contact momenten(1a), <b>aannames doen</b>
	Agile methode is nieuw voor teamleden.	DAD(8), tools(6a, 6b, 6c)
	OU heeft traag support	Github (6a)
	Geografische spreiding leidt tot communicatie problemen met kwaliteitsvermindering tot gevolg	skype(4), coordinatie(5), tools (6a, 6b)
	C++ is nog relatief nieuw voor 2 van de 3 programmeurs	leren(9), review(10)
	Het kost tijd om de bestaande verificatietools en de achterliggende xMAS -materie op te nemen.	tijd inplannen (11 en onderzoekscontext)
2	<i>Systeem kwaliteitsafname</i>	
	Agile is nieuw voor teamleden	agile (7), agile tool (6b)
	Structuurverval bij nieuwe features is een natuurlijk gevolg	Vaak refactoren (3), Architectuur(12)
	Geografische spreiding leidt tot communicatie problemen met kwaliteitsvermindering tot gevolg	skype(4), teamviewer(?), agile tool(6b)
	Ervaring met C++ beperkt met mogelijke gevolgen voor de kwaliteit (fouten, best practices missen)	leren(9), review(10)

verder op de volgende pagina

	risico	
	oorzaak	maatregel
3	<i>Product kwaliteitsafname</i>	
	Verificatie tools merken een fout niet op met consequenties voor in productie name van chips <sup>1</sup> .	testen en feedback users, eventueel beta versie
	<div style="border: 1px solid black; padding: 5px;"> <p><b>Aanpassing febr 2015</b></p> <p>Onvoldoende of trage feedback met als gevolg dat het eindproduct niet volledig naar de wensen van de opdrachtgever is.</p> </div>	<div style="border: 1px solid black; padding: 5px;"> <p><b>Aanpassing febr 2015</b></p> <p>Minstens 1 skype contactmoment per iteratie.</p> </div>

*Einde tabel relevante risico's*

Tabel 2: Maatregelen ter vermindering van risico's

	Maatregel	toelichting
id	Maatregel	Toelichting
1	<div style="border: 1px solid black; padding: 5px;"> <p><b>Aanpassing febr 2015</b></p> <p>Gepland overleg via skype, minstens 1x per iteratie</p> </div>	Dit verzekert een minimaal contact met opdrachtgever en begeleider
1a	vaste communicatiemomenten afspreken	Dit vermindert het effect van risico 1 (stakeholder beschikbaarheid)
2	Tussentijds contact via email	Dit vult de communicatietijd aan tot wat nodig is. Nadeel is een kans op vertraging.
3	Refactoring	Een refactor na uitvoeren en testen van een taak, levert een goed gestructureerd systeem na elke iteratie.
4	Skype en chat creatief en vaak gebruiken	Verhoogt de gelijkheid met lokaal samenwerken
5	Op afspraak tegelijkertijd bouwen	Verhoogt de kans om samen te werken
6	Ondersteuning zoeken buiten de OU	Een trage ondersteuning kan op kritieke momenten het gehele project vertragen. Door minder op OU support te leunen, vermijden we het risico
6a	Gebruik Github	Vermijdt risico traag OU support
6b	Gebruik agile tool	Vermindert risico van kwaliteitsafname
6c	Gebruik mailing list	Verbeterd communicatie over email
7	Agile literatuur lezen	Door ons actief in agile te verdiepen, verkleinen we de kans op problemen met het proces

*verder op de volgende pagina*

	Maatregel	toelichting
8	DAD methode met ondersteunend tool kiezen	DAD is beter geschikt voor mensen die minder ervaring met agile hebben. Het tool zorgt voor gemakkelijkere toepassing van DAD.
9	C++ studie doen	Door actief ons C++ 2011 eigen te maken, kunnen we met onze achtergrondkennis van programmeertalen en van Java, het risico op vertraging voor zijn
10	Review door de ervaren C++ programmeur	Learning on the job onder begeleiding van het teamlid dat C++ ervaring heeft
11	Voldoende tijd voor voorfase	De ontwikkeling pas starten na de domeinanalyse en het bepalen van de architectuur. Dit beperkt het gevaar van te vroeg implementeren
12	Architectuur	Vergroot de kans dat de structuur a priori geschikt is voor de applicatie en voor de nieuwe features.
13	<div style="border: 1px solid black; padding: 5px; margin-bottom: 5px;"> <b>Aanpassing febr 2015</b>  Aannames doen </div>	<div style="border: 1px solid black; padding: 5px;"> <b>Aanpassing febr 2015</b>  Voor kleinere zaken gaan we aannames doen. Alleen voor de duidelijk grotere zaken vragen we expliciet toestemming. Dit bevordert de voortgang. Oorzaak van deze maatregel is het ontbreken van onmiddellijke feedback (normaliter onderdeel van de agile aanpak). </div>

*Einde tabel maatregelen ter vermindering van risico's*



## 2.1 Proceskeuze

Bij de keuze tussen plangedreven, volledig agile en een hybride aanpak hebben we de factoren en overwegingen uit figuur 2 gebruikt.

<ol style="list-style-type: none"> <li>1. een geografisch gespreid team</li> <li>2. 8 maanden voor de uitvoering</li> <li>3. High level requirements bekend bij opdrachtgever en geen extern risico</li> <li>4. geen van de teamleden hebben ervaring met agile</li> <li>5. opdrachtgever en begeleider mogen een beperkte tijd besteden aan het project</li> <li>6. het team wil kennismaken met agile, maar niet ten koste van effectiviteit van het project</li> <li>7. de detailrequirements komen tijdens het bouwen aan de orde. Vooraf verzamelen is minder goed te realiseren</li> </ol>	<p><b>SDM2</b> een watervalmethode vergt rigide requirements. Hoewel de high level requirements bekend zijn, geldt dat niet voor detailrequirements. Deze zijn mede afhankelijk van het ontwerploop. Verder heeft een watervalaanpak een groter risico op uitloop doordat het team vooraf functionaliteit toezegt in een bepaalde tijd te realiseren. Ervaring wijst uit, dat uitloop vaker voorkomt dan niet.</p> <p>Ten slotte wenst het team ervaring met agile op te doen, voorzover dit binnen de doelstellingen van het project past.</p> <p><b>XP</b> Volledig agile is om meerdere redenen niet haalbaar (zie hieronder). Voor XP is de vrijwel continue beschikbaarheid van gebruikers noodzakelijk. Ook is pair programming niet uit te voeren met een geografisch gespreid team. Verder hebben de leden geen ervaring met agile projecten. Om deze redenen is het onverstandig een volledig agile proces op starten zoals XP.</p> <p><b>Hybride</b> Een hybride aanpak met cherry picking van methoden en technieken lijkt het meeste kans op succes te hebben.</p>
--	--

Figuur 2: Welk proces gaan we hanteren?

## Conclusies

XP valt af omdat het fysieke nabijheid vergt en grote gebruikers betrokkenheid

SDM2 heeft een te rigide requirements engineering proces en valt dus ook af.

Scrum valt eveneens af omdat het maar goed werkt als een team de nodige agile vaardigheden heeft. Verder is er ook een rol nodig van Scrum Master wat in een klein team al voor een onevenwicht zorgt in taakverdeling.

AUP (Agile Unified Process) is een vereenvoudigde versie van RUP, het sluit goed aan bij de visie van ons team maar wordt sinds 2006 niet meer ondersteund.

DAD (Disciplined Agile Delivery) is de verderzetting van AUP door Scott Ambler, DAD is een “people-first, learning-oriented hybrid agile approach to IT” . DAD is een raamwerk en de life-cycle bestaat uit drie fasen , de elaboratiefase wordt als een constructie gezien. In tegenstelling tot XP, Scrum en andere waar de focus voornamelijk op software ontwikkeling ligt, is bij DAD het ganse traject van belang. Overzicht waar DAD voor staat

1. People first : Self-disciplined, Self-organizing, Self-aware
2. Learning oriented : domain learning, process learning, technical learning.
3. Agile : enhances the values and principles of the Agile Manifesto.
4. Hybrid : adopts and tailors strategies from a variety of sources.
5. IT solution focused : provide real business value to your stakeholders.
6. Full delivery lifecycle : from the beginning of a project to the release of the solution into production.
7. Goals driven : focused on the right things at the right time
8. Risks and value driven : attack the risks before they attack you.
9. Enterprise aware

DAD lijkt voor ons als team en voor het project het meest geschikt, het stelt “learning oriented” voorop en houdt in dat domeinstudie en hoe je de stakeholders het best bedient even belangrijk is als het ontwikkelen van software. Het hanteert zachte mijlpalen en met de “Proven Architecture milestone” in het begin van de constructiefase zorgt men er voor dat de meeste risico’s rond architectuur geweken zijn eer men verder bouwt. Met de zogenaamde “Work Item” lijst worden te behandelen items volgens risico en waarde geprioriteerd.

# 3

## Architectuur

De architectuur levert een gestructureerde indeling van het systeem met als doel om enerzijds de software toegankelijk te maken voor ontwikkelaars en anderzijds input te leveren voor aanpassingen in de software. Hieronder een specificatie van de producten die deze activiteit oplevert.

Logical view	De logische structuur in class en object diagrammen.
Process view	De dynamische structuur in state transition diagrams op systeem niveau.
Physical view	De hardware componenten en de verdeling van software over de hardware componenten.
Guidelines & constraints	De richtlijnen voor bouw, test, herstructurering en documentatie.

Door het toepassen van DAD bekommt men op het einde van de inceptie fase een initiële visie op de architectuur. In het begin van de constructie wordt dan volgens een "risicowaarde" prioriteit de architectuur bepaald en bereikt men op de "Proven Architecture milestone" een architectuur die bruikbaar en getest is. Deze bepaalt de onderliggende componenten en hun interfaces. In principe wijzigt de architectuur niet meer, tenzij een nieuw requirement daar aanleiding toe geeft.

De domeinanalyse ligt op het domein van de klant: het geeft een deelprobleem. Met een domeinanalyse licht je één deel van het probleem uit het geheel en bestudeer je dat gedetailleerder en wetenschappelijker dan de andere deelproblemen.

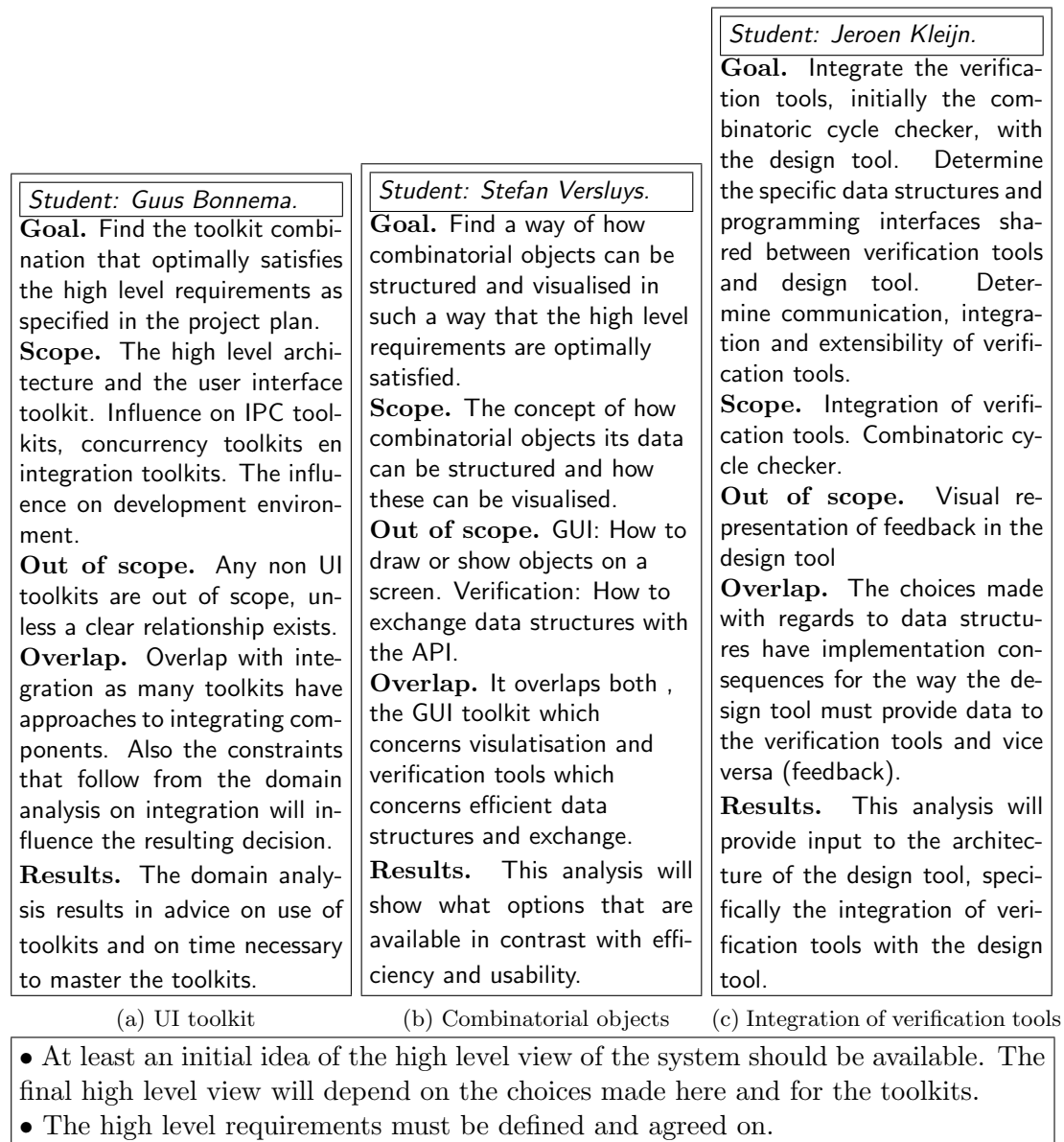
Ervaring wijst uit dat het belangrijk is de domeinanalyses als eerste te doen. Dat ondersteunt het project en voorkomt dat je de domeinanalyse aan het eind “nog even te moeten schrijven”.

De twee belangrijke kenmerken van een domeinanalyse zijn dat het een beperkt deel van het onderwerp is en dat het daadwerkelijk ondersteunend is aan het project. De valkuil is om een te breed onderwerp te nemen. Hieronder de voorbeelden van goede onderwerpen voor een domeinanalyse die voor ons als input hebben gediend.

De uitkomsten van een domeinanalyse zijn een beschrijving van het probleem, een beschrijving van de alternatieven, een beschrijving van de keuzecriteria en een gewogen aanbeveling. Gewogen betekent dat de aanbeveling naar objectieve criteria plaatsvindt. De beslissing is een team effort, waarbij de klant de doorslaggevende stem heeft. De uitwerking evenals de beoordeling van een domeinanalyse is individueel.

In het project kiezen we voor de grootste risico's eerst. Dit zijn met name de integratie en de user interface met ontwikkelomgeving. Hiermee halen we het grootste risico uit het project. Zie figuur 3 voor de uitwerking per persoon. De keuze van het team is voor Guus de UI toolkit, voor Jeroen de integratie (met combinatorische cycle checker) en voor Stefan de combinatorial objects.

vb	beschrijving
Platformonafhankelijk toolkit	Uitgaande van de high level requirements voor het ontwerptool, zoals platformonafhankelijkheid, uitbreidbaarheid en hechte integratie: kies een user interface toolkit die het beste past bij dit project en dit probleemgebied.
Integratie van verificatietools	Een analyse van de huidige en mogelijke toekomstige verificatietools. Welke datastructuren gebruiken zij intern en welke datastructuren kunnen de tools met elkaar delen? Hoe kunnen de verificatietools worden geïntegreerd met de ontwerptool? De combinatorische cycle checker is een goede basis om mee te beginnen.
Combinatorial objects	Een onderzoek hoe om te gaan met combinatorial objects. Bijvoorbeeld: hoe grafisch weer te geven, hoe op te slaan in de data structuur?



Figuur 3: Scope domain analyses

### 5.1 Introductie

Het project betreft het herbouwen van het WickedXmas tool, omdat de tool gebruikt wordt voor studie en ontwerp van chips is het belangrijk dat deze tool gemakkelijk en foutloos werkt op de gespecificeerde platformen. Verder betreft het een niet alledaags domein waardoor er extra aandacht noodzakelijk is om het domein te verkennen. De eis dat het tool platformonafhankelijk moet zijn en goed met de bestaande C++ programmatuur moet integreren, maakt dat tools noodzakelijk zijn. Er is tijd voorzien voor de teamleden om zich de ontwikkeltools eigen te maken. Wij hebben gekozen voor de agile aanpak DAD, het plan is opgebouwd volgens de drie fasen ervan.

### 5.2 Organisatie

De teamleden ontwikkelen gedelokaliseerd en communiceren via internet. Sources worden gecentraliseerd in de cloud met versiebeheer. Het team beschikt over een begeleider welke toeziet op de gang van zaken en geeft op vraag advies. De opdrachtgever en tevens domeinspecialist kan geraadpleegd worden voor specifieke domeinvragen en evaluatie van het product.

1. Guus Bonnema
  - Rol - Ontwikkelaar
  - Skills - IT, Java, Linux
2. Jeroen Kleijn
  - Rol - Ontwikkelaar
  - Skills - IT, MS VisualStudio (C#), C/C++, Java, Linux + Windows
3. Stefan Versluys
  - Rol - Ontwikkelaar
  - Skills - IT, MS VisualStudio (C#), C/C++, Java, Windows + VxWorks
4. Freek Verbeek
  - Rol - Proces begeleider
5. Bernard van Gastel
  - Rol - Opdrachtgever / Domeinspecialist : begeleidt inhoudelijk

Tijdens de constructiefasen komen er nog de DAD rollen van product owner en architecture owner bij. Wie die rollen gaan vervullen bepalen we tijdens de eerste twee iteraties. De DAD rol van team lead voorzien wij momenteel niet nodig te zullen hebben omdat het team daarvoor te klein is.

### 5.3 Hardware en software

1. Om versiebeheer en sources te borgen gebruiken wij een centrale Git repo.
2. Communicatiemiddelen zijn GitHub, gmail, skype, teamviewer en een tool om het agile werken te ondersteunen.
3. Platformonafhankelijke IDE voor het ontwerp tool met C++ compiler voor de analyse tools.
4. Platformonafhankelijke GUI Toolkit.
5. Componenten voor xMAS analyse en checks.
6. Mac OS, MS Windows, Linux platformen.
7. DAD Support Tool (Work Item list, Visualize work , Burn down chart)

### 5.4 DAD Ontwikkelmethode

#### 5.4.1 Lifecycle

Mijlpalen :

1. Stakeholder consensus
2. Proven architecture
3. Sufficient functionality
4. Production ready
5. Delighted stakeholders

#### 5.4.2 Iteratie aanpak

- Analyse : Modellen, requirements, prioriteiten en selectie
- (Test Driven) Development
- Refactoring
- Peer review
- Levert steeds iets bruikbaar op dat geëvalueerd kan worden voor feedback.
- Planning
- Documentatie

### 5.4.3 Contact momenten met opdrachtgever

Het ontwerp tool is voor de research naar chip ontwerpen van belang in verband met correctheid en toetsen van chip ontwerpen op globaal niveau, voordat de daadwerkelijke detail ontwerp en productie van chips begint. Voor de opdrachtgever is het ontwerp tool dermate belangrijk dat hij graag op de hoogte is van zowel wat we bouwen als hoe we het bouwen. In overleg met de opdrachtgever kiezen we daarom voor een relatief intens schema van contact. De opdrachtgever blijft zo goed in contact met wat we oplossen en hoe we het oplossen. Deze keuze stemt overeen met het karakter van agile ontwikkelen.

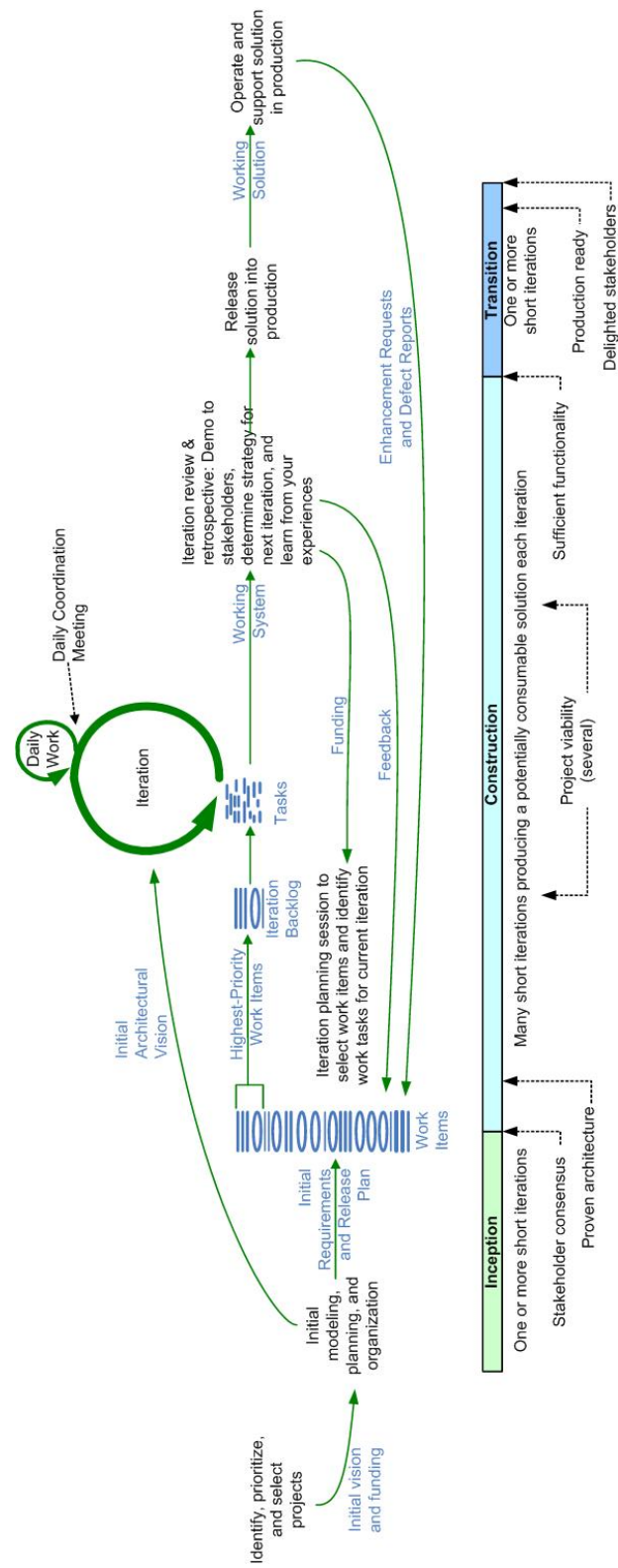
**Start iteratie** Aan het begin van de iteratie zit een fase van requirements ordenen, prioriteiten geven en selecteren voor implementatie. Tijdens deze fase speelt de opdrachtgever de hoofdrol.

**Voortgang** Midden in de bouw – b.v. na het ontwerp – informeren de teamleden de opdrachtgever over het ontwerp en de ontwerp beslissingen.

**Demo en evaluatie** Aan het einde van de iteratie tonen de teamleden het gebouwde onderdeel en evalueren het resultaat samen met de opdrachtgever.

**Ad hoc** Bij tussentijdse vragen of behoefte aan bevestiging kunnen de teamleden contact opnemen met de opdrachtgever.





Figuur 4: DAD basic Lifecycle

## 6.1 Indeling activiteiten over de drie DAD fasen

### 1. Inceptie (2 iteraties)

- Iteratie “Planning”
  - Team afspraken.
  - Stakeholder meeting.
  - Probleem definitie.
  - High level requirements.
  - Planning.

**Artefacten** De teamleden hebben de nodige afspraken gemaakt met betrekking tot werkwijzen zoals communicatiemiddelen en versiebeheer voor documentatie. Deze fase levert een document dat het probleem omschrijft, de high-level requirements, business case, risico's, stakeholders, vision and challenges, een plan van aanpak en een planning voor de eerstvolgende iteratie.

- Iteratie “Domeinanalyse”
  - Domeinanalyse.
  - Requirements: Op basis van sourcecode van de huidige WickedXmas tool.
  - Requirements: Demonstratie van de huidige WickedXmas tool door stakeholder.
  - Initiële architectuur visie

**Artefacten** De resultaten van de domeinanalyse en requirements op basis van observatie. Er is een visie m.b.t. de initiële architectuur. Documentatie.

### 2. Constructie (6 iteraties + 2 keer onderzoekscontext)

- Iteratie 0
  - Architectuur : Afbakenen , configureren en testen
  - documentatie aanpassen

**Artefacten op de “Proven architecture” mijlpaal** De Architectuur is bepaald en getest en klaar voor evaluatie, documentatie.

- Onderzoekscontext
- Iteratie 1
  - WickedXmas editor ontwikkelen
  - documentatie aanpassen

**Artefacten** Prototype, documentatie

- Onderzoekscontext
- Iteratie 2
  - WickedXmas editor ontwikkelen
  - documentatie aanpassen

**Artefacten** Prototype, documentatie

- Iteratie 3
  - WickedXmas editor ontwikkelen
  - documentatie aanpassen

**Artefacten** Prototype, documentatie

- Iteratie 4
  - WickedXmas verificatietoolinterface ontwikkelen
  - documentatie aanpassen

**Artefacten** Prototype, documentatie

- Iteratie 5
  - WickedXmas verificatietoolinterface ontwikkelen
  - documentatie aanpassen

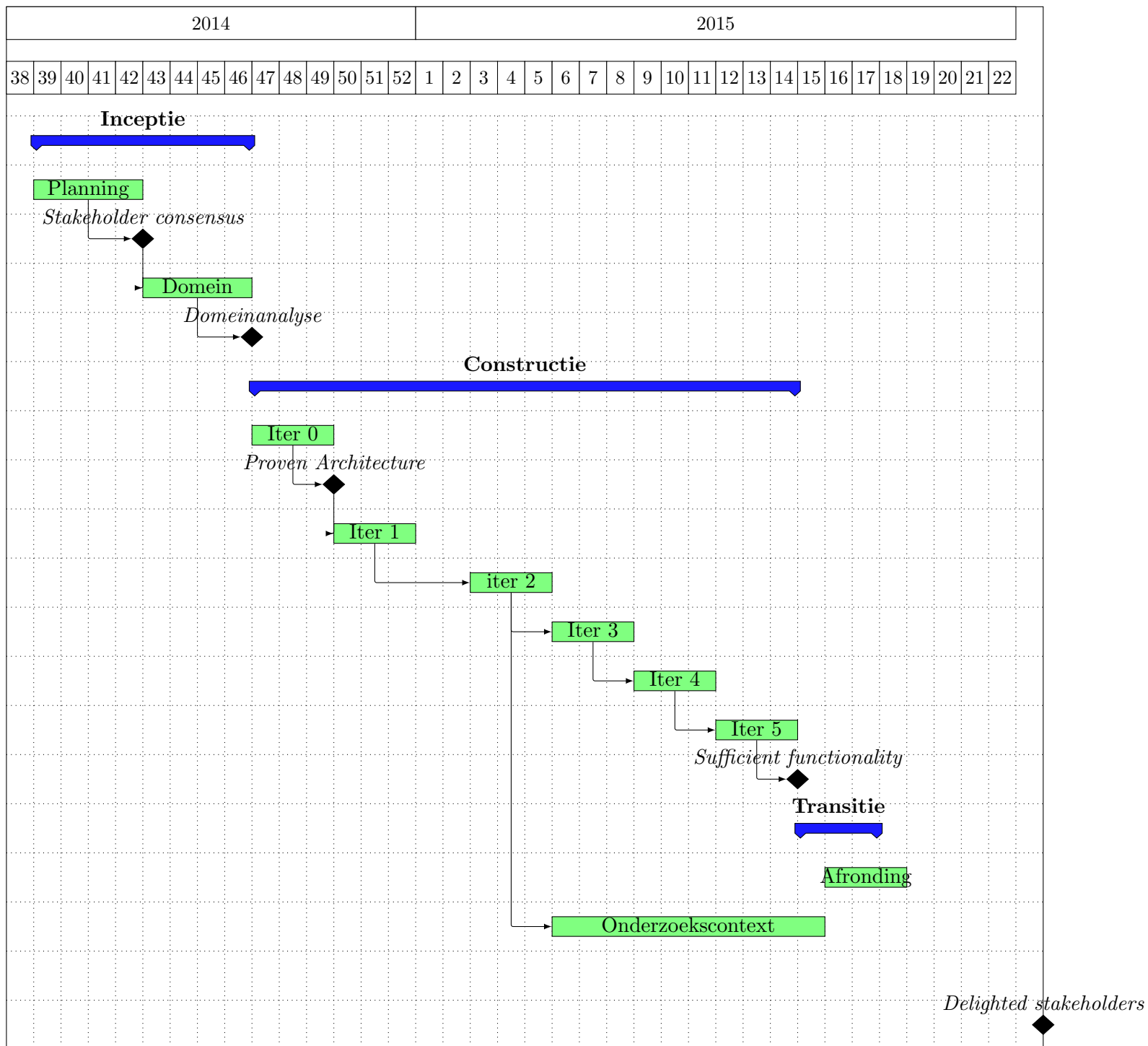
**Artefacten op de “Sufficient functionality” mijlpaal** Een release van de WickedXmas Tool zoals beoogd werd, documentatie

### 3. Finale transitie (1 iteratie)

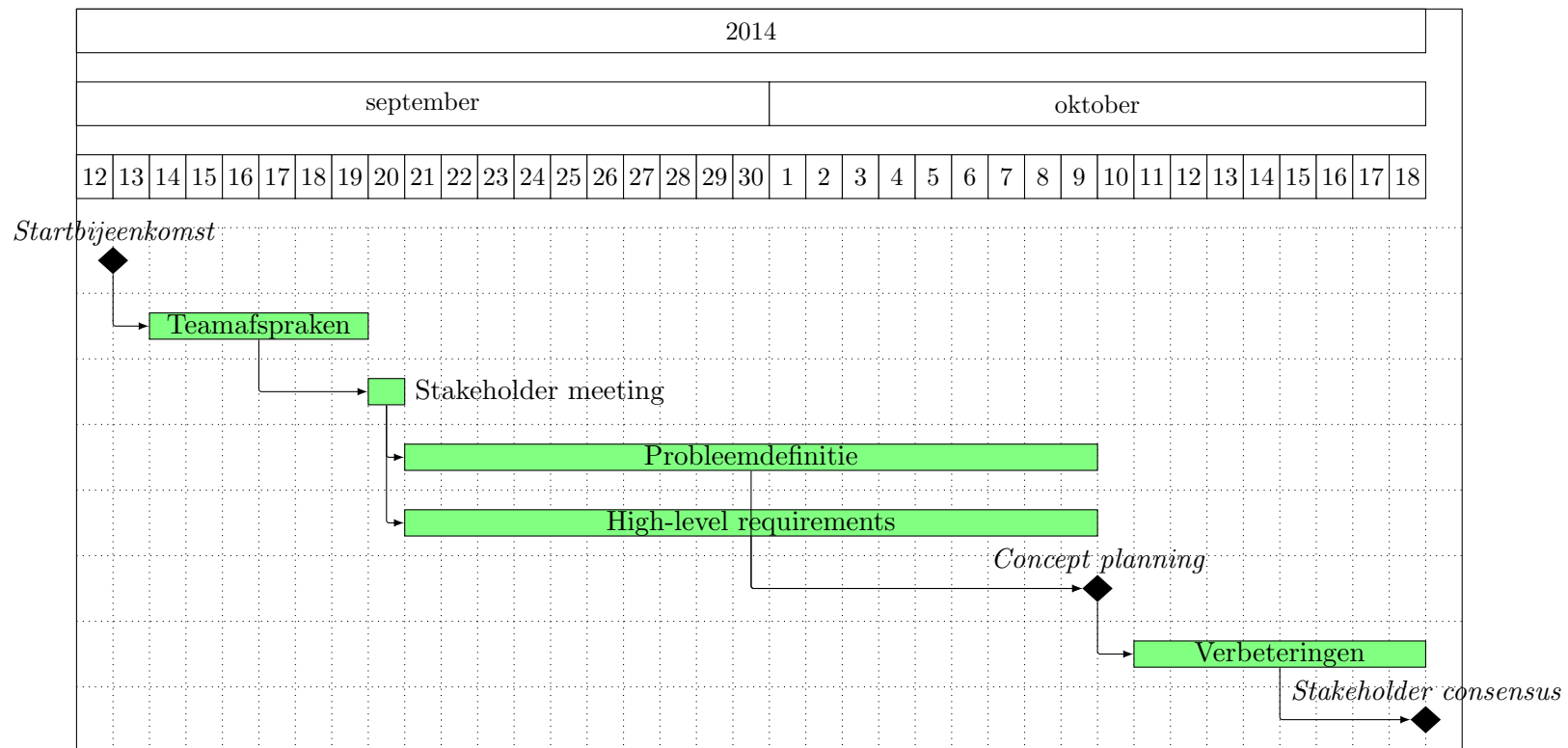
- Finale releases van de nieuwe WickedXmas tool.
- Documentatie bundelen (handleiding).
- Onderzoekscontext
- Presentatie geven

**Artefacten op de “Production ready” mijlpaal** Een presentatie, documentatie en scriptieverslag van de nieuwe WickedXmas tool.

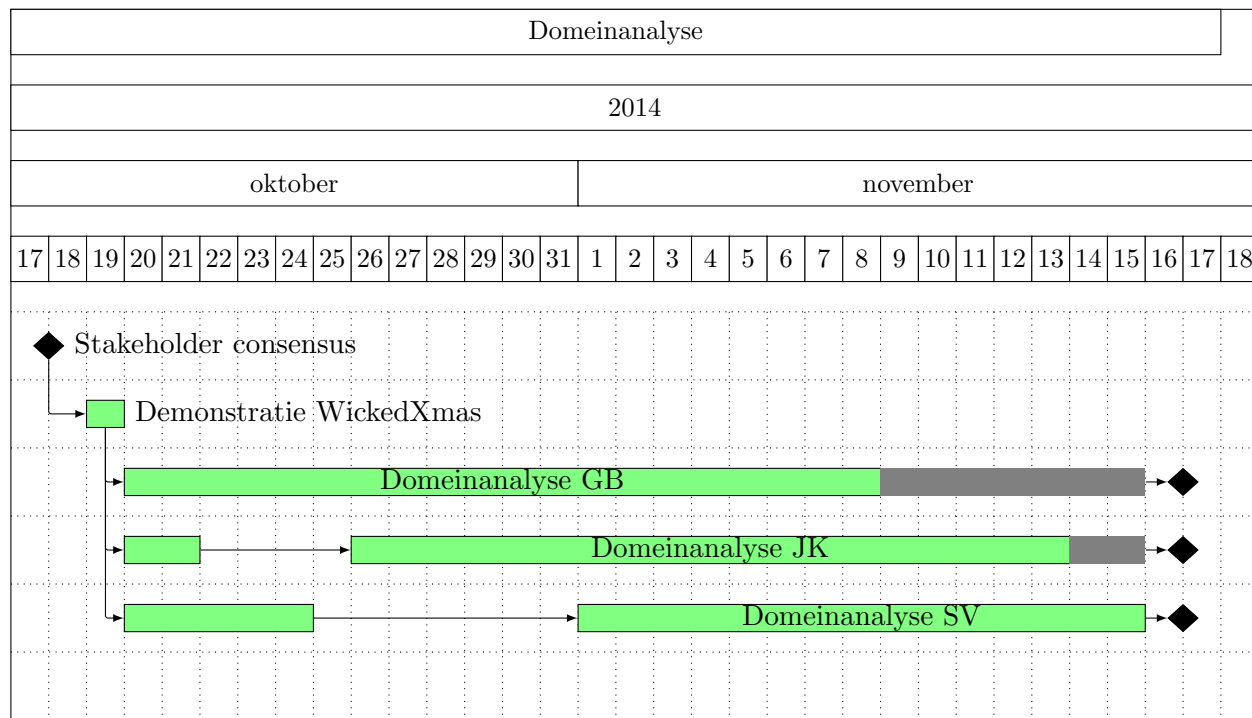
## 7 | Schedule



Figuur 5: Globale planning met hoofdfasen en belangrijkste mijlpalen



Figuur 6: Detailplanning 'Planning'



Figuur 7: Detailplanning 'Domeinanalyse'

Fase	weken
Planning	2,5
Domeinanalyse	3
Iteratie 0	3
Iteratie 1	3
Iteratie 2	3
Iteratie 3	3
Iteratie 4	3
Iteratie 5	3
Onderzoekscontext	2
Afronding	1.5
Totaal	27

Figuur 8: Studielast taken

- looptijd project: tot eind mei (36 weken)
  - 27 weken effectief voor het project ( $27 \text{ weken} * 15 \text{ uur/week} = 405 \text{ uur}$ )
  - 3 weken vakantie
  - 6 weken speling
- De detailplanning van een fase wordt telkens in de voorafgaande fase uitgewerkt.

### Planning iteraties

Elke iteratie omvat een vaste periode van 3 weken. De exacte verdeling van de beschikbare tijd over de taken verschilt per iteratie. In de eerste iteraties zal bijvoorbeeld relatief meer tijd worden besteed aan requirements. Aan het einde van een iteratie wordt een voorlopige release van de software aan de opdrachtgever verstrekt. De laatste twee dagen zijn gereserveerd voor het werken aan het scriptieverslag en de onderzoekscontext. In deze periode vindt ook een evaluatie plaats van de afgelopen iteratie.

Voor het vaststellen van de requirements zal regelmatig overleg met Bernard plaats moeten vinden. Hiervoor is het belangrijk dat in de detailplanning van een iteratie rekening wordt gehouden met de beschikbaarheid van Bernard.

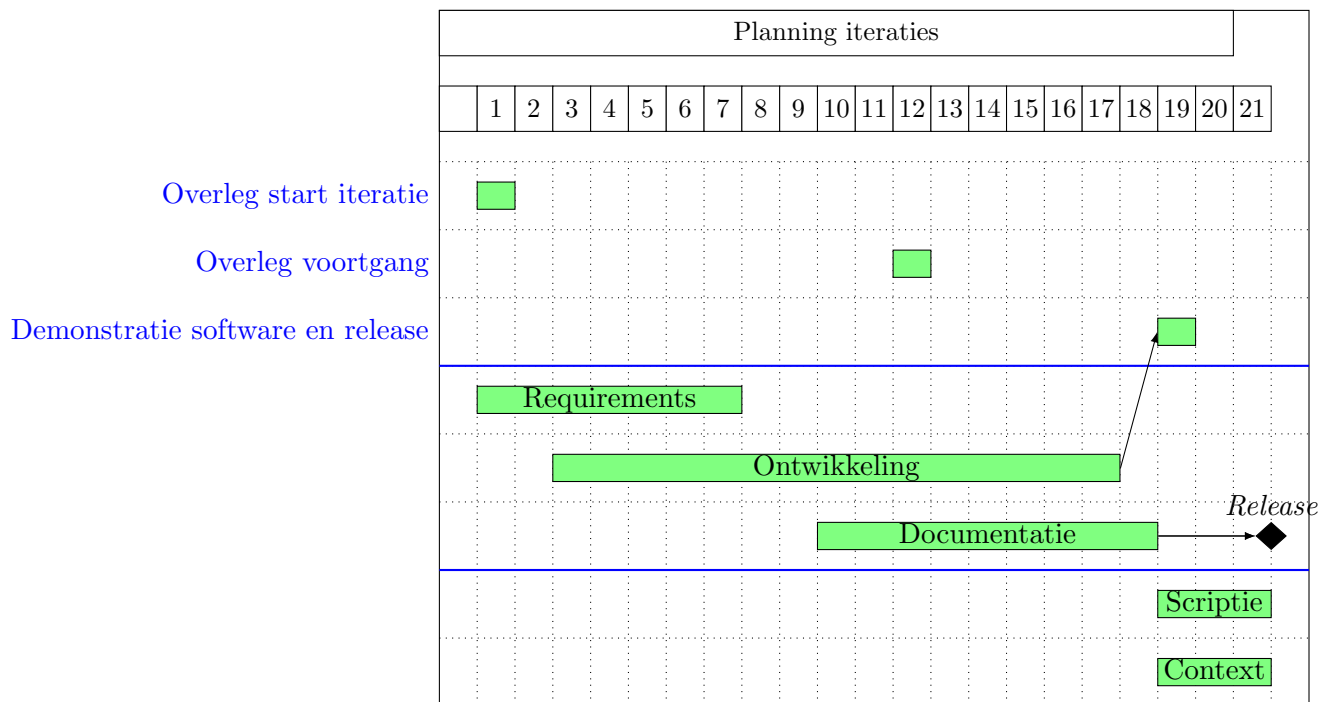


Week	Taak	Toelichting
38-42	Planning	
42-45	Domeinanalyse	4 weken ingepland i.v.m. vakanties Stefan & Jeroen
46-48	Iteratie 0	
49-51	Iteratie 1	
52-1		vakantieperiode, mogelijk individueel (bijvoorbeeld onderzoekscontext)
2-4	Iteratie 2	
5-7	Iteratie 3	
8-10	Iteratie 4	
11-13	Iteratie 5	
5-14	Onderzoekcontext	Uitvoering parallel aan de iteraties
15-16	Afronding	
22	Mijlpaal einde project	

Figuur 9: Planning taken

## Capaciteitsplanning

- Beschikbare tijd
  - $\sim 15$  uur per week per teamlid
  - $\sim 30$  uur gedurende het hele project voor Freek en  $\sim 20$  uur voor Bernard gereserveerd vanuit het ABI. Freek en Bernard hebben echter aangegeven dat zij graag intensief betrokken worden tijdens het project. Zij zullen dus meer tijd willen en kunnen vrijmaken. Afgesproken is dat wij op elk moment contact op kunnen nemen. Zodra wij te veel van hun tijd in beslag nemen zullen zij dat aangeven.
- Vakanties / niet beschikbaar voor project
  - Stefan
    - \* 25-10-2014 t/m 31-01-2014 (week 44)
    - \* 24-12-2014 t/m 02-01-2015 (week 52-01)
  - Jeroen
    - \* 21-10-2014 t/m 25-10-2014 (week 43)
  - Bernard
    - \* 04-10-2014 t/m 14-10-2014 (week 41-42)
    - \* 20-10-2014 t/m 27-10-2014 (week 43-44)



Figuur 10: Globale planning iteraties

## Referenties

- [CKO12] Satrajit Chatterjee, Michael Kishinevsky, and Umit Y Ogras. xmas: Quick formal modelling of communication fabrics to enable verification. *Journal of IEEE Design & Test of Computers*, Aug 2012.

## A | Definities

**analyse module** synoniem met verificatie module

**i18n** Internationalization, referring to translation of menu items, system documentation etc.

**l10n** Localization, referring to country specific settings such as money, numbers, dates etc.