Bijlage III

PLAN VAN AANPAK

De koppeling van een (L)CMS aan hoogwaardige en dynamische E-learning modules

tinqwise

Auteur: Willianne Timmer
Opdrachtgever: TinQwise
Bedrijfsbegeleider: Dhr. H. Van Os
Stagebegeleider: Dhr. S. van der Linden
Datum: 14 maart 2007

Inhoudsopgave

Inhoudsopgave **2**
1. Probleemstelling **3**
1.1 Probleem 3
1.2 Onderzoeksvragen 3
1.2.1 Centrale vraag 3
1.2.2 Deelvragen 3
1.3 Doelstelling 3
2. TinQwise **4**
2.1 Zakelijke informatie 4
2.2 Geschiedenis 4
2.3 Algemene informatie 4
2.3.1 Visie en missie 4
2.3.2 Organisatiestructuur 5
2.4 Markten 6
3. Het proces **7**
3.1 Plan van aanpak 7
3.2 Analyse fase 7
3.3 Design fase 8
3.4 Realisatie fase 8
3.5 Implementatie fase 8
3.6 Scriptie schrijven 8
3.7 Eindpresentatie 8
4. Grenzen **9**
4.1 Algemeen 9
4.2 Uitgangspunten en randvoorwaarden 9
5. Planning **10**

1. Probleemstelling
1.1 Probleem

TinQwise ontwikkelt E-learning applicaties. Dit gebeurt op verschillende manieren, met verschillende eisen. Zo zijn sommige applicaties gemaakt om online gebruikt te worden en andere zijn juist weer voor offline gebruik. Soms moeten applicaties makkelijk geüpdatet kunnen worden, maar in andere gevallen is die mogelijkheid overbodig. Ook verschillen de doelen per applicatie. Elke E-learning applicatie is anders en vraagt om maatwerk. Er wordt dus ook gebruik gemaakt van verschillende CMS'en, LMS'en en LCMS'en. De applicaties worden daarnaast in verschillende programma's gebouwd, denk hierbij aan bijvoorbeeld Lectora of Flash, maar ook Eedo en zelfgebouwde tools. Deze probleemstelling is toegespitst op de E-learning modules die dynamisch (zie deelvraag 3) zijn en daarom in bepaalde programma's (zoals Flash) gebouwd worden. Tot nu toe is het nog niet gelukt om dit aan een (L)CMS te koppelen. Voordelen hiervan (makkelijk kunnen updaten van modules zou een voordeel kunnen zijn) kunnen meerwaarde kunnen hebben voor TinQwise.

Tijdens deze afstudeerstage zal daarom onderzocht worden of een CMS of LCMS kan worden gekoppeld aan dynamische E-learning modules.

1.2 Onderzoeksvragen
1.2.1 Centrale vraag

Hoe kunnen de voordelen van een (L)CMS gekoppeld worden aan hoogwaardige en dynamische E-learning modules?

1.2.2 Deelvragen

1. Welke meerwaarde bieden (L)CMS'en voor E-learning modules?

2. Wat zijn de kenmerken van E-learning modules (Hierbij beperkt dit onderzoek zich tot de modules die geproduceerd zijn bij TinQwise gedurende de afgelopen twee jaar)?

3. Wat maakt E-learning modules dynamisch?

1.3 Doelstelling

Het doel van deze afstudeeropdracht is om een advies te geven aan TinQwise over het koppelen van een (L)CMS aan E-learning modules. Hierbij wordt ook advies gegeven over welk (L)CMS het beste gekoppeld kan worden en op welke manier.
Daarnaast zal een prototype ontwikkeld worden. Dit zal gebeuren naar aanleiding van dit onderzoek.

2. TinQwise
2.1 Zakelijke informatie

Stagebedrijf:
Tinqwise
Olympia 3d – 1
1213 NS Hilversum

T 035 – 538 56 56
E tinqwise@tinqwise.nl

Stagebegeleider:
Hans van Os
Functie: Projectleider
M 06-15016488
E hansvanos@tinqwise.nl

2.2 Geschiedenis

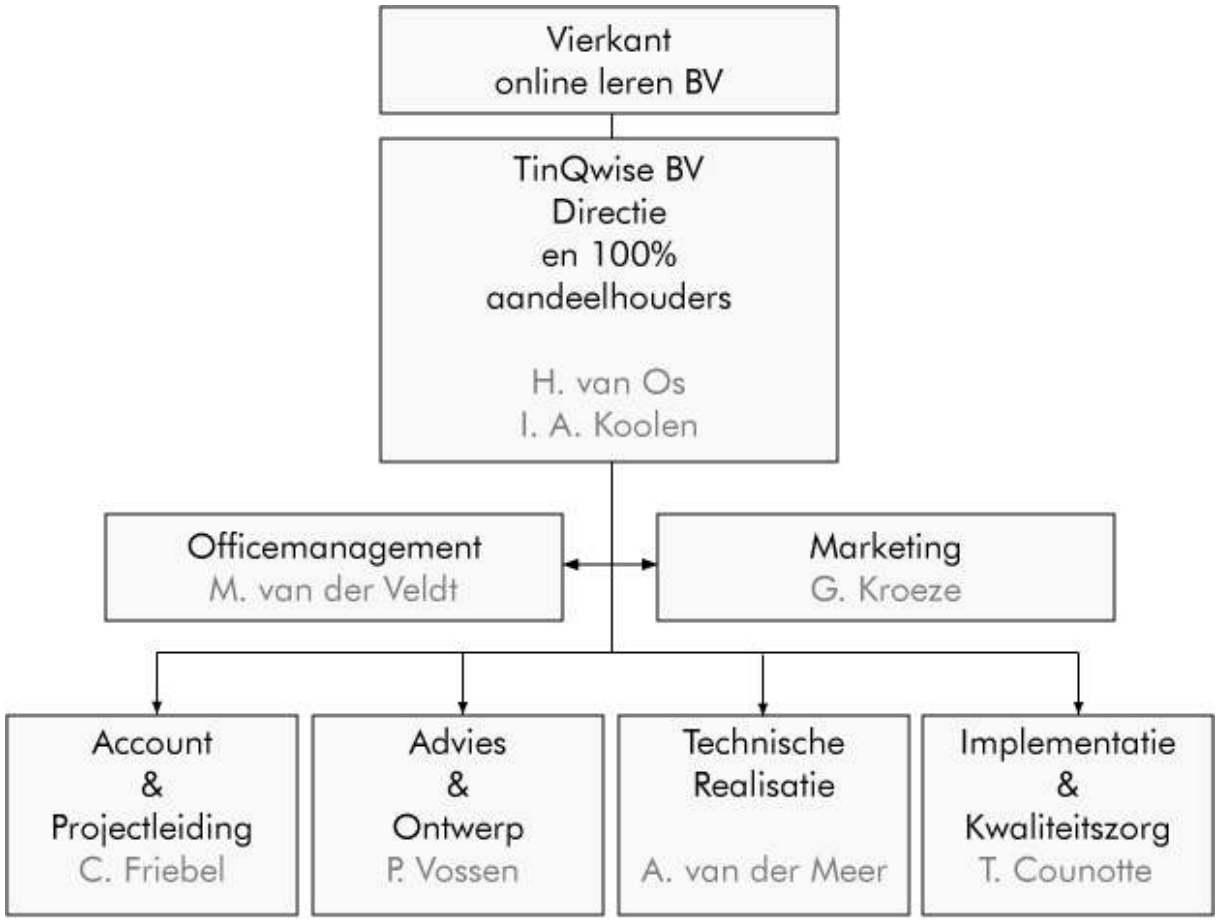
TinQwise is in 1985 opgericht als een onderneming gespecialiseerd in het ontwerpen en produceren van interactieve mediaproducties. Niet onder de naam TinQwise, maar als de stichting Film en Wetenschap. Daarmee is het in Nederland één van de oudste bureaus dat gespecialiseerd is in interactieve media. TinQwise is een bedrijf dat zich voornamelijk richt op E-learning. Sinds haar oprichting ontwikkelt TinQwise opleidingsprogramma's waarbij gebruik wordt gemaakt van de combinatie van computertechnologie, audiovisuele media en netwerkmedia zoals Internet en Intranet. Dit alles doet TinQwise vanuit haar thuisbasis in Hilversum.

2.3 Algemene informatie
2.3.1 Visie en missie

Naar de mening van TinQwise moet E-learning alleen worden ingezet als het een meerwaarde heeft. Een voorbeeld van een kwalitatieve meerwaarde is dat het leerrendement wordt verhoogd. Een voorbeeld van een kwantitatieve meerwaarde is bijvoorbeeld het reduceren van de kosten. In beide gevallen zijn de organisatie-doelstellingen het uitgangspunt en niet het middel. De oplossingen van TinQwise voor vraagstukken van opdrachtgevers omvatten daarom vaak veel meer dan alleen de ontwikkeling van een E-learning toepassing. TinQwise kijkt bij een opdracht graag naar twee aspecten: het leerrendement én de inpassing in de organisatie.

2.3.2 Organisatiestructuur

De organisatiestructuur van TinQwise is op transparante wijze ingericht op het ontwikkelen van E-learning toepassingen. TinQwise heeft een flexibele, platte organisatie. Op dit moment heeft het bedrijf 25 medewerkers in dienst.



Figuur 1: Organogram TinQwise

TinQwise is een full service onderwijskundig en multimediabureau dat alle disciplines in huis heeft. Zo heeft TinQwise, naast creatieve teams en softwarespecialisten, deskundigheid op het gebied van audiovisuele media, opleidingsorganisatie en ontwikkeling, interne en externe communicatie en de nieuwste digitale mediatechnieken.

TinQwise kent een matrixachtige structuur. Per opdracht(gever) kan zeer snel een multidisciplinair team worden ingericht. Afhankelijk van het karakter van de opdracht worden een communicatie adviseur, een projectleider, een vormgever, een scenarioschrijver en een software specialist in een team bij elkaar gebracht. Vaak wordt een dergelijk team aangevuld met één of meer deskundigen van de opdrachtgever.

2.4 Markten

TinQwise werkt voor een groot aantal grote organisaties en ondernemingen, zowel binnen als buiten de overheid. Kenmerkend is dat TinQwise langdurende relaties met haar opdrachtgevers aangaat. Voorbeelden van deze oprachtgevers zijn het Ministerie van Defensie, Albert Heijn, O&O Fonds, KOC Nederland (Verzorgingsbranche) en ING Group.

De klanten die TinQwise heeft, zijn lastig in te delen in verschillende groepen. De klanten opereren vaak in verschillende sectoren. om de klanten toch binnen een kader te kunnen plaatsen is een grof onderscheid te maken:

- bedrijfsleven
- brancheverenigingen
- onderwijs
- overheid

Dat de methode die TinQwise hanteert werkt, blijkt uit het feit dat een groot aantal ondernemingen en organisaties al vele jaren naar tevredenheid met TinQwise samenwerkt.

3. Het proces

In dit hoofdstuk worden alle activiteiten die tijdens de afstudeeropdracht worden doorlopen, opgenomen. Elke activiteit is opgedeeld in een individueel proces om tot het desbetreffende eindproduct te komen.

3.1 Plan van aanpak

- Gesprekken met de bedrijfsbegeleider en stagebegeleider
- Inhoud vaststellen
- Opleveren afstudeerovereenkomst
- Afstudeerovereenkomst doortrekken naar het plan van aanpak

3.2 Analyse fase

- In kaart brengen van de huidige situatie bij TinQwise
  - -Gesprek met bedrijfsbegeleider
  - -Gebruik maken van al bestaande documenten
- Welke meerwaarde bieden (L)CMS'en voor E-learning modules?
  - -Uitzoeken wat de kenmerken zijn van een (L)CMS in het algemeen
  - -Uitzoeken welke (L)CMS'en gebruikt worden door navragen
  - -Uitzoeken wat de kenmerken zijn van deze (L)CMS'en door navraag en internetonderzoek
  - -Uitzoeken waarom keuzes zijn gemaakt door navraag en interview
- Wat zijn de kenmerken van E-learning modules?
  - -Hoe is de technische infrastructuur en waarom? Verdiepen in reeds gemaakte E-learning modules en navraag
  - -Welke keuzes met betrekking tot didactiek zijn er gemaakt en waarom?
  - -Interviews met didactici
  - -Verwachtingen door navraag
- Wat maakt E-learning modules dynamisch?
  - -Interviews met technici en didactici
  - -Definitie
  - -Omschrijvingen
- Technische verkenning
  - -Welke technieken zijn benodigd om het einddoel te bereiken
- Conclusies trekken
  - -Hoofd- en deelvragen beantwoorden
  - -Formuleren van eisenlijst voor de uiteindelijke applicatie
  - -Onderzoeksverslag voltooien

3.3 Design fase

• Ontwerp voor het prototype maken aan de hand van de eisenlijst en de technische verkenning. Eventueel de technische verkenning doorzetten waar nodig
• Plan van aanpak om het prototype te realiseren, overleggen met technici wat mogelijk is

3.4 Realisatie fase

• Realisatie van prototype (nader onder te verdelen in stappen)
• Evaluatie

3.5 Implementatie fase

• Test van het prototype
• Als er tijd is een verbeterde versie van het prototype
• Voldoet het product? Vragen, interviews

3.6 Scriptie schrijven

• Inhoud, inleiding schrijven
• Eerste hoofdstukken schrijven aan de hand van afstudeerovereekomst
• Onderzoeksverslag verwerken
• Motivatie, werkwijze, methode bespreken
• Ontwerp bespreken
• Ontwerpproces bespreken
• Implementatie bespreken
• Evaluatie
• Samenvatting schrijven
• Lay-out van de scriptie

3.7 Eindpresentatie

• Presentatie maken in Powerpoint
• Voorbereiden (materialen in orde etc.)
• Presenteren

4. Grenzen
4.1 Algemeen

Begindatum: 1 februari 2007
Einddatum: 1 juli 2007

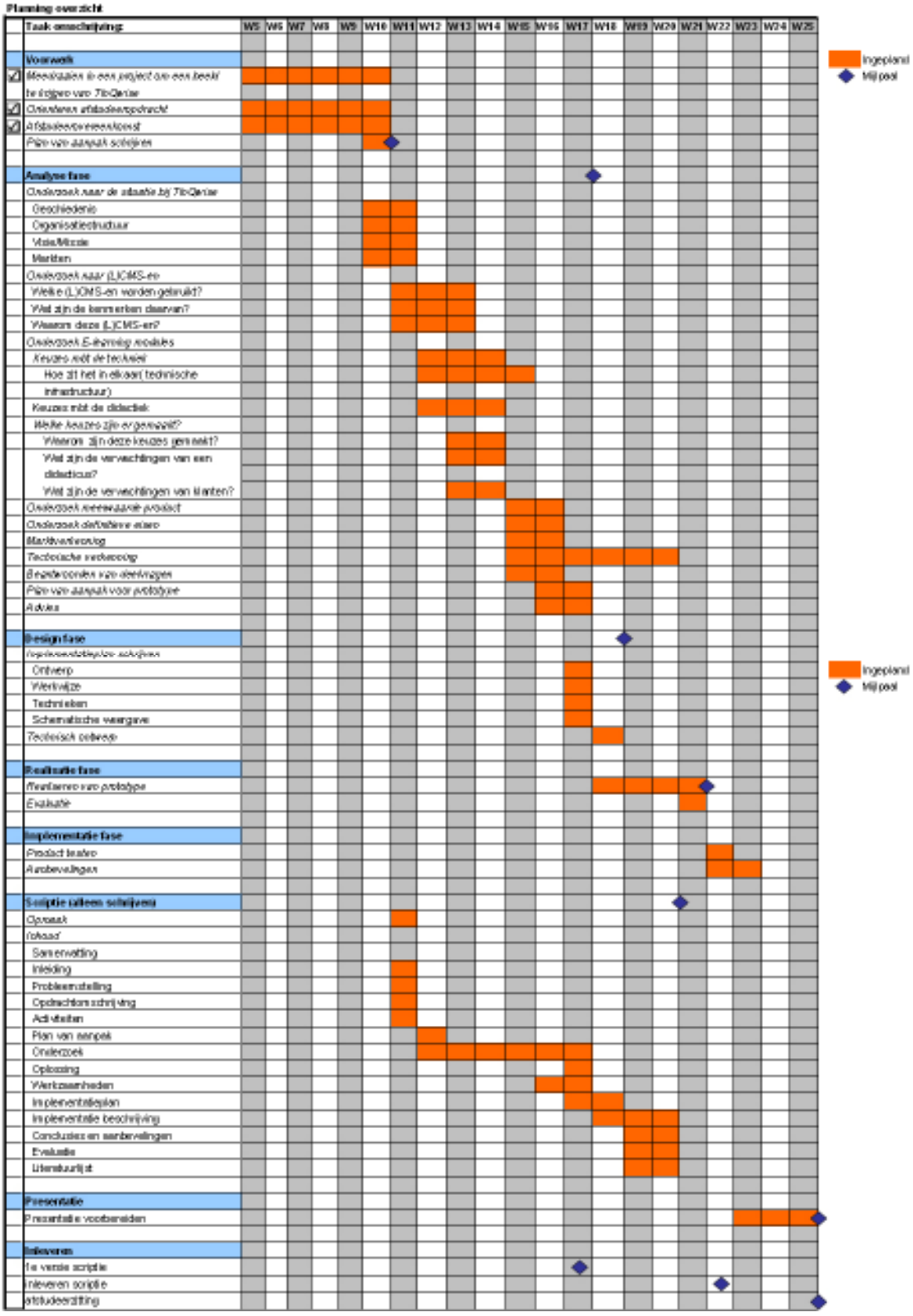
Totaaltijd: 840 uur

4.2 Uitgangspunten en randvoorwaarden

TinQwise ontwikkelt E-learning applicaties. Dit gebeurt op verschillende manieren, met verschillende eisen. Zo zijn sommige applicaties gemaakt om online gebruikt te worden en anderen zijn juist weer voor offline gebruik. Soms moeten applicaties makkelijk geüpdatet kunnen worden, maar in andere gevallen is die mogelijkheid overbodig. Ook verschillen de doelen per applicatie. Elke E-learning applicatie is anders en vraagt om maatwerk. Er wordt dus ook gebruik gemaakt van verschillende CMS'en, LMS'en en LCMS'en. De applicaties worden daarnaast in verschillende programma's gebouwd, denk hierbij aan bijvoorbeeld Lectora of Flash, maar ook Eedo en zelfgebouwde tools. De probleemstelling is toegespitst op de E-learning modules die dynamisch (zie deelvraag 3) zijn en daarom in bepaalde programma's (als Flash) gebouwd worden. Tot nu toe is het nog niet gelukt om dit aan een (L)CMS te koppelen. Voordelen hiervan (makkelijk kunnen updaten van modules zou een voordeel kunnen zijn) kunnen meerwaarde kunnen hebben voor TinQwise.

Afbakening
-Het onderzoek wordt tot de (L)CMSen die gebruikt worden binnen TinQwise beperkt.
-Het onderzoek wordt beperkt tot de E-learning modules die ontwikkeld worden en ontwikkeld zijn binnen TinQwise in de afgelopen twee jaar.

5. Planning


Planning overzicht
Taak omschrijving
W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15 W16 W17 W18 W19 W20 W21 W22 W23 W24 W25
Voorwerk
Meedraaien in een project om een beeld te krijgen van TicQwise
Oriënteren afstudeeropdracht
Afstudeerovereenkomst
Plan van aanpak schrijven
Analyse fase
Onderzoek naar de situatie bij TicQwise
Geschiedenis
Organisatiestructuur
Visie/Missie
Markten
Onderzoek naar (L)CMS-en
Welke (L)CMS-en worden gebruikt?
Wat zijn de kenmerken daarvan?
Waarom deze (L)CMS-en?
Onderzoek E-learning modules
Keuzes mbt de techniek
Hoe zit het in elkaar(technische infrastructuur)
Keuzes mbt de didactiek
Welke keuzes zijn er gemaakt?
Waarom zijn deze keuzes gemaakt?
Wat zijn de verwachtingen van een didacticus?
Wat zijn de verwachtingen van klanten?
Onderzoek meewaarde product
Onderzoek definitieve eisen
Marktverkenning
Technische verkenning
Beantwoorden van deelvragen
Plan van aanpak voor prototype
Advies
Design fase
Implementatieplan schrijven
Ontwerp
Werkwijze
Technieken
Schematische weergave
Technisch ontwerp
Realisatie fase
Realiseren van prototype
Evaluatie
Implementatie fase
Product testen
Aanbevelingen
Scriptie (alleen schrijven)
Opmaak
Inhoud
Samenvatting
Inleiding
Probleemstelling
Opdrachtomschrijving
Activiteiten
Plan van aanpak
Onderzoek
Oplossing
Werkzaamheden
Implementatieplan
Implementatie beschrijving
Conclusies en aanbevelingen
Evaluatie
Literatuurlijst
Presentatie
Presentatie voorbereiden
Inleveren
1e versie scriptie
inleveren scriptie
afstudeerzitting
Ingepland
Mijlpaal
Ingepland
Mijlpaal

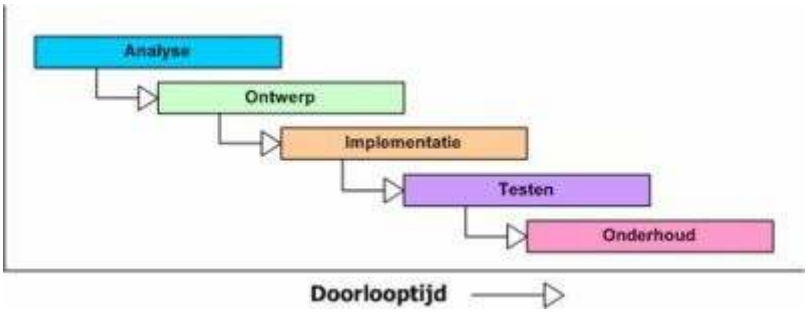
Bijlage II

Keuze voor de onderzoeksmethode

Er zijn verschillende gangbare ontwikkelmethodes voor het (object georienteerd) ontwikkelen van software. Een methode waar tijdens het keuze-semester HCI mee gewerkt werd, was het 'Designing the user experience'-model van UPA. In dit model worden vier verschillende fasen gedefinieerd: de analyse fase, de design fase, de implementatie fase en de deployment fase. Dit model is voor gericht op user-interface design. Om deze reden is gezocht naar een methode die iets beter bij dit afstudeerproject aansluit.

Er wordt gebruik gemaakt van diverse modellen. Een zoektocht levert het volgende lijstje op:
-Waterval model
-Fountain model
-Iteratief model
-Incrementeel model
-Spiraal model

Een klassieke ontwerpmethode is de waterval methode. In de waterval methode volgen de fasen elkaar lineair op en iedere volgende fase gaat uit van de resultaten uit de vorige fase. Nadeel hiervan is dat er geen iteratie mogelijk is. Op de waterval methode zijn diverse varianten, waaronder het 'Aorta-lifecycle' model en het 'Sashimi' model. Bij het 'Sashimi' model zit het verschil met het traditionele waterval model in het feit dat de fasen elkaar overlappen, zodat terugval in fasen tot op zekere hoogte mogelijk is.



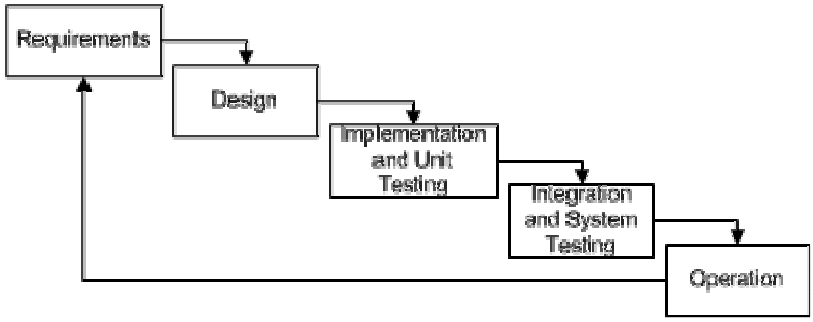
Figuur 1: Variant op de waterval methode, het 'Sashimi' model waarin de opeenvolgende fasen elkaar overlappen.

Het fountain model is gebaseerd op het waterval model. Bij het fountain model volgen de fasen elkaar ook op een lineaire manier op. Hier is echter wel enige iteratie mogelijk tussen de fasen.

Een ander model is het iteratieve model (iterative application development, IAD). Dit is een iteratieve ontwikkel methode, wat inhoudt dat een ontwerp-lus steeds opnieuw doorlopen kan worden.

Figuur 2: Model volgens IAD

Het incrementeel model is gebaseerd op het waterval model. Het model onderscheidt verschillende fasen opeenvolgend, net als het waterval model, alleen is er na deze cyclus een iteratie mogelijk. Wel is het zo dat na de eerste cyclus al een werkend prototype beschikbaar is. Het incrementeel model is flexibeler dan het waterval model.

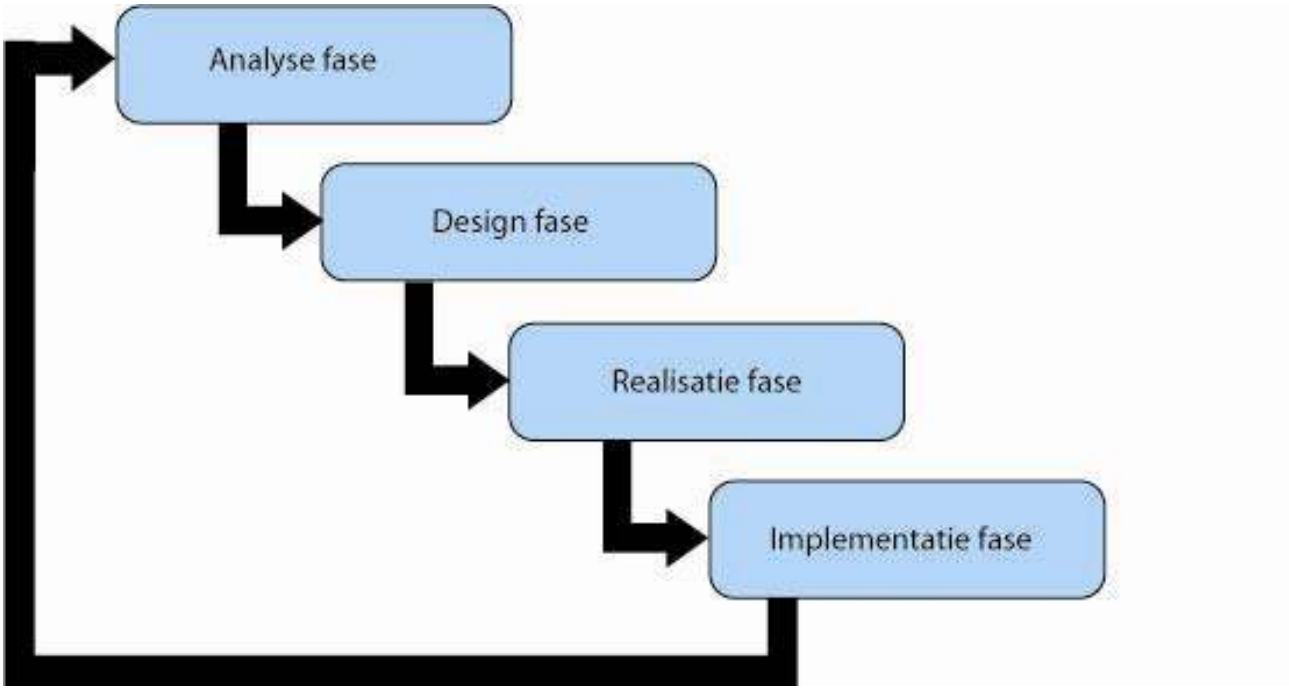


Figuur 3: Voorbeeld van het incrementeel model

Het spiraal model is gebaseerd op iteraties. Het onderscheidt verschillende fasen, planning, risico analyse, techniek en evaluatie. In een spiraal doorloopt het de verschillende fasen.

Uit bovenstaande modellen moet uiteindelijk een keuze gemaakt worden. Het incrementeel model spreekt het meest aan. Er zal een eigen schema gemaakt worden waarin de te doorlopen fasen gedefinieerd zullen worden. Ook wordt er gekozen om nog enige overlap tussen de opeenvolgende fasen te houden, dus eigenlijk wordt de methode die gebruikt gaat worden een combinatie tussen het incrementeel model en het 'Sashimi' model. Tijdens deze afstudeeropdracht in ieder geval één lus doorlopen worden, zodat het project na het afstudeerproject eventueel voortgezet kan worden.

Het model dat gebruikt gaat worden, ziet er als volgt uit:



Figuur 4: Uiteindelijk gekozen onderzoeksmodel

Bijlage III

Handleiding voor het gebruik van dynamische Flash-objecten in Eedo ForceTen

Willianne Timmer
25/06/2007

Inhoud:

Inleiding 2

Flash voorbereiden voor dynamisch gebruik 3
Flash-ontwikkelaar

Het invoeren van de dynamische Flash-content 4
Basis content-beheerder

Het exporteren van de cursus 5
Gevorderde content-beheerder

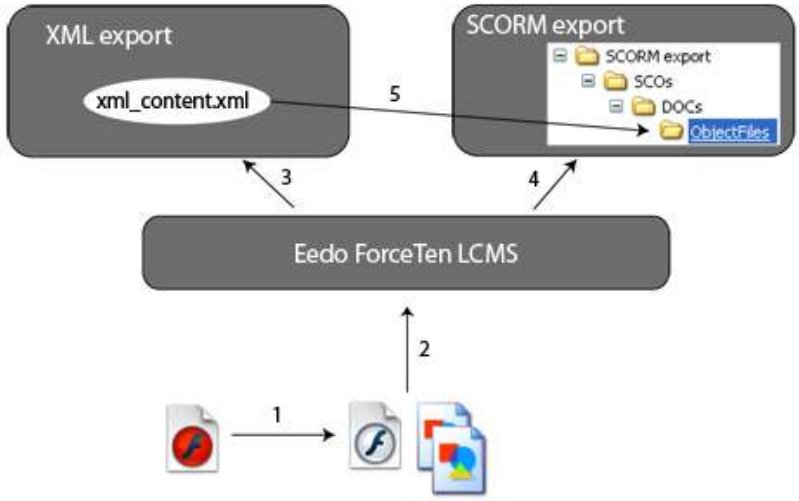
Bijlage – ActionScript 6

Inleiding

TinQwise ontwikkelt E-learning applicaties. Dit gebeurt op verschillende manieren en met verschillende eisen. Een belangrijk onderdeel van een E-learning applicatie is interactie met de gebruiker. Met een programma als Flash kunnen interactieve elementen aan een E-learning applicatie worden toegevoegd.

In deze handleiding wordt beschreven hoe Flash-bestanden gemaakt kunnen worden waarin in Eedo ForceTen dynamische content kan worden toegevoegd. Dit proces is hieronder ook schematisch weergegeven. Deze handleiding is geschreven voor zowel de Flash ontwikkelaar als de content-beheerder. Omdat er in Eedo ForceTen vaak meerdere content-beheerders zijn is er onderscheid gemaakt tussen de basis-beheerder en de gevorderde content-beheerder.

Het gebruik van dynamische Flash applicaties in Eedo ForceTen



1. Maak een swf bestand dat variabelen haalt uit xml_content.xml
2. Zet een .swf bestand in Eedo met bijbehorende variabelen in text- en image objecten
3. Exporteer een XML package
4. Exporteer een SCORM package
5. Zet xml_content.xml in de map objectFiles in het SCORM package

Handleiding voor het gebruik van dynamische Flash-objecten in Eedo ForceTen

Flash voorbereiden voor dynamisch gebruik

Belangrijk: Er is hier uitgegaan van al aanwezige basiskennis van de Flash ontwikkelaar betreffende het werken in Flash met ActionScript. Een voorbeeldscript (DFEscript.as) wordt bijgeleverd, de inhoud is te vinden in de bijlage. Er mag hier natuurlijk aan gesleuteld worden, zolang de basis gelijk blijft.

Subdoel:
Het Flash bestand opbouwen met externe variabelen uit xml_content.xml en externe afbeeldingen.

Vooraf:
Zorg dat je weet welke teksten en afbeeldingen extern zijn. Maak in Excel een lijstje met een kolom 'type' (graphic of text), een kolom 'variabele naam' en een kolom 'inhoud' en vul deze velden in voor alle nodige variabelen. Het kan zijn dat de inhoud van de tekstvelden nog niet bekend is. Zorg in dat geval dat de daarvoor verantwoordelijke persoon dit op tijd doet.
Zorg ook dat je weet of er een default variabele in Flash staat, zodat tekstvakken of afbeelding placeholders nooit leeg hoeven te zijn. Overleg dit met de contentschrijver.

Werkwijze:

1. Opbouw van de applicatie

-Werk met dynamische tekstvakken.
-Zorg dat de naamgeving voor jezelf duidelijk is.
-Als je werkt met afbeeldingen, maak dan placeholders, movieclips met een default uiterlijk. Deze afbeeldingen kunnen dan later gewisseld worden.

2. Variabelen toevoegen met ActionScript

-Zorg dat het bestand xml_content.xml ingelezen wordt in frame 1. Zorg dat je het XML-bestand laadt uit de map ObjectFiles. Het pad wordt dus: ObjectFiles/xml_content.xml.
-De variabelen die wijzen op de inhoud zitten altijd op een vast niveau, op een willekeurige plek in de xml. De inhoud van de variabelen zit ook altijd op een vast niveau. Om bij de juiste inhoud te komen, moet dus gezocht worden naar de opgegeven variabele namen, bijvoorbeeld flashtekstvak1 of flashafbeelding1.
-Zorg dat de variabelen op de juiste plekken terechtkomen, het maakt niet uit welke extensie een afbeelding heeft, dit wordt vanzelf opgepakt. Waar wel op gelet moet worden, is dat de afbeelding uit de map ObjectFiles komt. Hier moet dus naar verwezen worden.

Zie verder de ActionScript file voor de ActionScript, er zit commentaar bij.

Handleiding voor het gebruik van dynamische Flash-objecten in Eedo ForceTen

Het invoeren van de dynamische Flash-content

Belangrijk: Er is hier uitgegaan van al aanwezige basiskennis van de gebruiker betreffende het werken in Eedo ForceTen. De gebruiker kan al pagina's en objecten aanmaken in Eedo ForceTen. Er wordt hier alleen ingegaan op het dynamische Flash-object.

Subdoel:
Het Flash-bestand inclusief variabelen en afbeeldingen in Eedo ForceTen plaatsen.

Vooraf:
Zorg dat je alle beschikbare bestanden en variabelen klaar hebt staan. Als het goed is, staan de bestanden in de map van je project onder CdImage/Assets/Flash Eedo. Je hebt de volgende bestanden:
-Het Flash-bestand (vb: flashbestand.swf). De afmetingen van het Flash-bestand moeten ook al bekend zijn.
-Eventuele afbeeldingen (vb: afbeelding1.jpg, afbeelding2.gif)
-Een Excel-document waarin 3 kolommen staan met variabele namen, type variabelen en inhoud van de variabelen.

Werkwijze:

1. Het plaatsen van het Flash-bestand in Eedo ForceTen

-Zorg dat je werkt in de pagina waar het Flash swf bestand moet komen.
-Kies onderin de balk voor New, kies daarna bij de iconen voor Flash.
-Kies het juiste bestand (bijvoorbeeld flashbestand.swf) en stel alles in zoals het gewenst is. Belangrijk is de juiste width en height (in pixels). Dit zijn de afmetingen fan het Flash-bestand. Vergeet niet op Save te drukken.

2. Het plaatsen van de variabelen in Eedo ForceTen

-Werk het rijtje af uit het Excel document. Het type variabele vind je in de kolom: type. Dit is meteen het soort object dat je nieuw moet aanmaken in Eedo, een tekst object of een afbeelding. Hieronder staat een uitleg per type.
Tekstvariabelen:
-Kies onderin de balk voor New, kies daarna bij de iconen voor Text.
-De naam (Title) is heel belangrijk: geef de naam uit de tweede kolom in de Excel sheet (de kolom variabele naam). Vul dit in in het vak Title. Zorg dat hier geen fouten in staan!
-Vul in het tekst object zelf de tekst in uit de derde kolom van de Excel sheet (de kolom inhoud). Vergeet niet op te slaan.
-De tekst verschijnt in de linkerbovenhoek. Sleep de tekst op het Flash-object in je pagina. Dubbelklik op het teksts-object, druk op de knop 'Send To Back.'
Afbeeldingen:
-Kies onderin de balk voor New, kies daarna bij de iconen voor Graphic. Blader en voeg de afbeelding in.
-De naam (Title) is heel belangrijk: geef de naam uit de tweede kolom in de Excel sheet (de kolom variabele naam). Vul dit in in het vak Title. Zorg dat hier geen fouten in staan! Vergeet het object niet op te slaan.
-De afbeelding verschijnt in de pagina. Sleep de tekst op het Flash object in je pagina. Dubbelklik op de afbeelding, druk op de knop 'Send To Back'

3. Pagina opslaan

-Druk in de Eedo pagina op Save
-Sla de structuur op

Handleiding voor het gebruik van dynamische Flash-objecten in Eedo ForceTen

Het exporteren van de cursus

Doel:
Bij het exporteren van de applicatie moet gezord worden dat het juiste xml-bestand (waar de variabelen in staan) in de juiste map met het Flash bestand (waar de variabelen in ingelezen moeten worden) moet worden geplaatst.

Werkwijze:

1. XML Exporteren

-Ga met de menubalk naar Content > Export > XML.
-Zoek de cursus op die geëxporteerd moet worden, exporteer deze
-Sla de XML-export op de gewenste locatie op, dit is een Zip-file.
-Unzip het bestand
-Alleen het bestand xml_content.xml is nodig, de rest mag verwijderd worden.

2. De applicatie exporteren

-Ga met de menubalk naar Content > Export > Course > SCORM.
-Zoek de cursus op die geëxporteerd moet worden, kies de gewenste SCORM versie en kies 'finish'.
- Sla de SCORM-export op de gewenste locatie op, dit is een Zip-file.
-Pak de Zip-file uit er verschijnen nu een aantal bestanden en een map SCOs.

3. Het xml-bestand toevoegen aan de SCORM-export

-Zet het bestand xml_content.xml uit de XML-export in de map SCOs/DOCs/ObjectFiles

Bijlage

Actionscript (zie ook DFEscript.as)

```
/*lees de xml in*/
xmlGegevens = new XML();
xmlGegevens.ignoreWhite = true;
xmlGegevens.onLoad = leesXML;
//gebruik het bestand uit de map ObjectFiles, ook al staat de .swf hierin.
Het bestand waar de .swf ingeladen wordt staat namelijk een niveau hoger.
xmlGegevens.load("ObjectFiles/xml_content.xml");
/*deze functie wordt uitgevoerd als de xml geladen is*/
function leesXML(geladen) {
/*waarde is de variabele die veranderd als de juiste variabelen gevonden
zijn, dan is de while loop afgelopen*/
      var waarde = "nul";
/*in i en j wordt het niveau opgeslagen waarin de juiste variabelen zich
bevinden*/
      var 1c1 = -1;
      var 2c1 = -1;
      var 1c2 = -1;
      var 2c2 = -1;
      var 1c3 = -1;
      var 2c3 = -1;
/*node en node2 bevatten de inhoud van de objectnamen uit Eedo, deze worden
in de while loop vergeleken totdat er een match is tussen de naam en het
onderdeel*/
      var node =
xmlGegevens.firstChild.childNodes[0].childNodes[1c1].childNodes[1c2].childN
odes[1c3].firstChild.nodeValue;
      var node2 =
xmlGegevens.firstChild.childNodes[0].childNodes[2c1].childNodes[2c2].childN
odes[2c3].firstChild.nodeValue;
/*in de while loop wordt gezocht naar het niveau van de variabelen in de
xml totdat de waarden bekend zijn*/
      while (a<10 && waarde == "nul") {
            1c1++;
            2c1++;
            1c2 = -1;
            2c2 = -1;
            while (1c1<10){
                  1c1++;
                  2c1++;
                  1c3 = -1;
                  2c3 = -1;
                  while (1c3<10){
                        1c3++;
                        2c3++;
            node =
xmlGegevens.firstChild.childNodes[0].childNodes[1c1].childNodes[1c2].childN
odes[1c3].firstChild.nodeValue;
            node2 =
xmlGegevens.firstChild.childNodes[0].childNodes[2c1].childNodes[2c2].childN
odes[2c3].firstChild.nodeValue;
/*als de juiste node gevonden is wordt er een variabele gemaakt (tekst1,
afbeelding1) waarin het niveau opgeslagen wordt*/
                  if (node == "flashtekstvak1") {
                        waarde1 = "ok";
                        var niv1c1 = 1c1;
                        var niv1c2 = 1c2;
```

```
//tel 2 bij k op omdat de inhoud van de gevonden variabele (first child)
third child is.
                        var niv1c3 = 1c3 + 2;
                  }
                  if (node2 == "flashafbeelding1") {
                        waarde2 = "ok";
                        var niv2c1 = 2c1;
                        var niv2c2 = 2c2;
                        var niv2c3 = 2c3 + 2;
                  }
/*als er aan deze voorwaarde wordt voldaan wordt waarde iets anders, en
eindigt de while loop*/
            if (node == "flashtekstvak1" && node2 == "flashafbeelding1") {
                  waarde = "ok";
            }
      }
/*als de variabelen gevonden zijn worden de tekstvakken ingevuld met de
tekst uit het juiste niveau uit de xml, zijn de waarden niet aanwezig
verschijnt er een default tekst*/
      if (geladen) {
            if (waarde1 == "ok") {
                  _root.dynamischTekstVeld.text =
xmlGegevens.firstChild.childNodes[0].childNodes[niv1c1].childNodes[1c2].chi
ldNodes[niv1c3].firstChild.nodeValue;
            } else {
                  _root.dynamischTekstVeld.text = "Staat niet in de XML";
            }
/*ook wordt de juiste afbeelding door middel van de naam uit de xml gehaald
en flash leest hem vervolgens uit de map uit. De movieclip
'afbeeldingplaceholder' plaatst de afbeelding*/
            if (waarde2 == "ok") {
                  var var2 =
xmlGegevens.firstChild.childNodes[0].childNodes[niv2c1].childNodes[niv2c2].
childNodes[niv2c3].firstChild.nodeValue;
                  loadMovie(["ObjectFiles/"+var2.toString()],
"afbeeldingplaceholder");
            } else {
                  afbeeldingplaceholder.attachMovie("undef", undef,
this.getNextHighestDepth());
            }
      }
}
stop();
```