

Introductie tot de cursus

- 1 Inleiding 7
- 2 Leerdoelen 11
- 3 Voorkennis 12
- 4 Cursusmateriaal 12
- 5 Studeeraanwijzingen 12
- 6 Begeleiding en tentaminering 13



Introductie tot de cursus

1 Inleiding

In deze cursus wordt een overzicht gegeven van enkele aspecten die van belang zijn voor het succesvol uitvoeren van softwareprojecten. Elk softwareproduct kent een eigen levenscyclus, de Software Life Cycle (SLC) of vaak ook Software Development Life Cycle (SDLC) genoemd. Een definitie van een SLC is (IEEE Standard Glossary of Software Engineering Terminology, 1983):

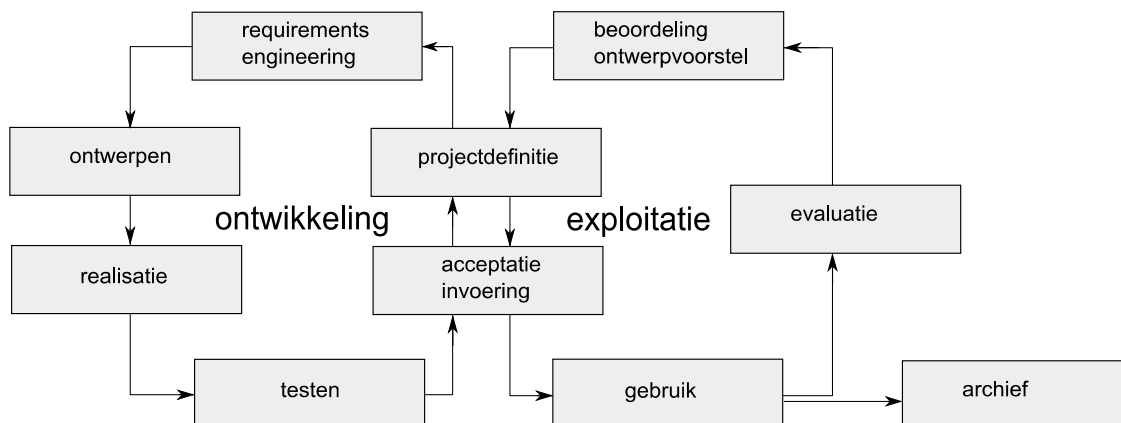
'The period of time that starts when a software product is conceived and ends when the product is no longer available for use. The software life-cycle typically includes a requirements phase, design phase, implementation phase, test phase, installation and checkout phase, operation and maintenance phase and sometimes, retirement phase'.

Voor het uitvoeren van softwareprojecten zijn in de loop van de tijd verschillende modellen voor SLC's ontwikkeld, de software life cycle modellen (SLCM). Deze beschrijven de volgorde en de wijze waarop een aantal activiteiten uitgevoerd zou moeten worden om te komen tot een succesvol softwareproduct.

Elk SLC-model heeft eigen uitgangspunten en bij toepassing in de praktijk specifieke voor- en nadelen. Er zijn vele SLC-modellen in gebruik, al is in vrijwel elke situatie sprake van een mengvorm van een of meer modellen. Elk project, product en organisatie 'vragen' om een eigen invulling van een SLC-model. Kennis over belangrijke aspecten van SLC-modellen is dus belangrijk om in de praktijk bij een concreet product een geschikt model te kunnen kiezen.

Het is echter ondoenlijk om alle modellen, aspecten en fasen die van belang kunnen zijn in een cursus als deze te behandelen¹.

Wat wij in deze cursus aan de orde stellen is echter meer dan wat normaliter met SLC-modellen wordt bedoeld. We bekijken niet alleen de modellen zelf maar gaan ook verder in op de processen die binnen en buiten het softwareproces werkzaam zijn. Hiervoor kijken we naar een algemeen model voor de interne processen bij de ontwikkeling en exploitatie van ict-systemen. Zie figuur 1.



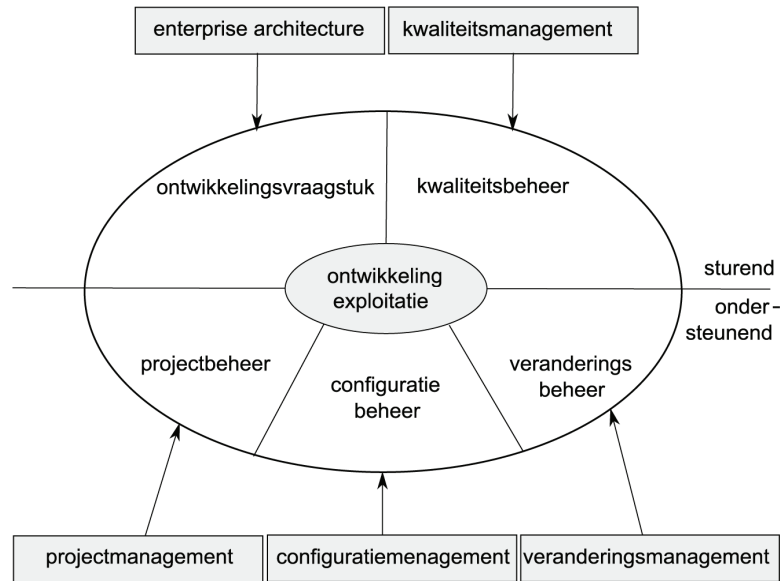
FIGUUR 1 Bicyclusmodel, interne processen

Dit model kenmerkt zich door twee in elkaar grijpende cycli (de ontwikkelingscyclus en de exploitatiecyclus) en wordt daarom ook wel het bicyclusmodel genoemd. De cycli bestaan uit processen die, wanneer ze sequentieel worden uitgevoerd overeenkomen met fasen in de ontwikkeling.

De ontwikkelingscyclus start met de projectdefinitie. Op grond hiervan volgen de fasen van requirements engineering, het ontwerpen, de realisatie en het testen van het systeem. Hierna volgt de acceptatie van het systeem en gaat dit in exploitatie. Het kan voorkomen dat een systeem niet geaccepteerd wordt. Dan volgt er, als er genoeg middelen beschikbaar zijn, een hernieuwde ontwikkelingscyclus om het systeem te verbeteren.

In de exploitatiecyclus zal het ict-systeem mogelijk na verloop van tijd geëvalueerd worden en komt er een verbetervoorstel (soms ook van IST naar SOLL-situatie genoemd) en een nieuwe projectdefinitie. Dit kan leiden tot een nieuwe ontwikkelingscyclus voor het systeem. Het kan natuurlijk ook zijn dat het systeem om diverse redenen niet meer voldoet en dat er een compleet nieuw systeem ontwikkeld moet worden. Het huidige systeem raakt dan in onbruik (archieff).

De bicyclus wordt aangestuurd door een aantal externe processen zoals weergegeven in figuur 2. Een intern proces kunnen we zien als de manifestatie van een extern proces bij een specifiek project. De externe processen kunnen nog verdeeld worden in inhoudelijk sturende processen en projectondersteunende processen. In figuur 2 is dit schematisch weergegeven door de horizontale lijn.



FIGUUR 2 Bicyclusmodel, interne en externe processen

In deze cursus kijken we in leereenheid 1 eerst naar een aantal SLC-modellen in het algemeen.

Daarna benaderen we het softwareproces vanuit de projectondersteunende processen. Eerst behandelen we in leereenheid 2 aspecten die te maken hebben met het leiden van ict-projecten, dus projectmanagement. Een actueel onderwerp op dit gebied is SCRUM, een moderne 'agile' projectmanagementmethode. Dit komt in leereenheid 3 aan de orde.

Een logisch vervolg hierop is leereenheid 4, waarin we kijken naar hoe deze (projectmanagement)processen verbeterd kunnen worden. Een organisatie kan hierdoor komen tot een hoger kwaliteitsniveau van producten en diensten, en zich zo profileren als een 'professionele organisatie'. In figuur 2 valt dit onder veranderingsmanagement.

Daarna benaderen we het softwareproces vanuit de inhoudelijk sturende processen. In leereenheid 5 komt kwaliteitsmanagement aan de orde. Het is daarbij belangrijk een goed beeld te hebben van softwarekwaliteit. Certificering op kwaliteit van een bedrijf kan een belangrijke reden zijn voor klanten om met een bedrijf in zee te gaan.

De leereenheden 1 tot en met 5 van de cursus zijn gebaseerd op belangrijke originele publicaties in het vakgebied. Hierin worden vooral de algemene principes beschreven die in de praktijk van industriële softwareprojecten daadwerkelijk tot verbeteringen hebben geleid. De cursus beoogt dat u door zelfstandig zoeken en verwerken van wetenschappelijke literatuur kennis en inzicht in het vakgebied verwerft. Het gaat er daarbij vooral om dat u onderscheid maakt tussen hoofd- en bijzaken, tussen sterke en zwakke uitspraken, tussen bereikt succes en verbloemd falen. Ook het vormen van een persoonlijke visie wordt van groot belang geacht.

Vanuit de optiek van inhoudelijk sturende processen komen we bij het bepalen van de enterprise architectuur bij het interne proces van eisenspecificatie, onderdeel van het gehele ontwikkelingsvraagstuk. Het opstellen en beheer van de eisen valt onder 'requirements engineering'. Dit proces komt uitgebreid aan de orde in deze cursus in leereenheid 6 en verder.

Uit een recent literatuuronderzoekⁱⁱ naar karakteristieken voor het kiezen van het juiste SLCM blijkt dat de invloed van een aantal voor de hand liggende aspecten (zoals requirements) van het softwareproces eigenlijk nauwelijks onderzocht is. De aard van de requirements en de mate waarin requirements door ontwikkelaars worden begrepen kan bepalend zijn bij de keuze voor een SLCM. Hieruit mag blijken dat alle genoemde processen, gezichtspunten en de onderlinge relaties van belang zijn voor het begrijpen van de interne en externe processen tijdens de SLC.

De leereenheden 1 tot en met 5 worden afgesloten met het maken van een kort verslag waarin u een verband weet te leggen tussen de eigen werksituatie en de verworven kennis omtrent softwaremanagement uit deze leereenheden. Zie hiervoor ook de informatie hieronder bij de paragraaf *Tentaminering*.

De leereenheden 6 t/m 11 over Requirements Engineering zijn gebaseerd op het boek 'Requirements Engineering, From System goals to UML models to Software specifications' van Axel van Lamsweerde, John Wiley, 2009.

De leereenheden 6 en 7 geven aan de hand van het tekstboek een overzicht van het vakgebied en van de courante methoden en technieken die gebruikt worden op het terrein van requirements elicitation. Aansluitend volgt een hoofdstuk (leereenheid 8) over de rol die doelen (goals) kunnen spelen in requirements engineering. Deze leereenheid vormt een belangrijke brug naar leereenheid 9; de daar behandelde methode is goal-oriented. Deze hoofdstukken geven vooral een breed overzicht van bestaande methoden en technieken; elk daarvan wordt kort en daardoor noodzakelijkerwijs oppervlakkig behandeld. Het doel van dit deel is om een overzicht te geven van de stand van het vakgebied. Er wordt van u verwacht dat u de behandelde methoden en technieken kunt plaatsen en dat u de zwakke en sterke kanten ervan weet. Er wordt echter niet verwacht dat u deze methoden en technieken volledig zelfstandig kunt toepassen. De nadruk ligt dus op weten, niet op kunnen.



Leereenheid 9 en 10 behandelen enkele essentiële onderdelen van de KAOS-methode voor het opstellen van requirements. Er wordt een leereenheid besteed aan:

- het opstellen van een goal model
- het maken van een risicoanalyse en het uitbreiden van het goal model met behulp van obstacle analyse.

Leereenheid 11 bevat nog een oefencasus die in de online sessies aan bod komt. Dit ter ondersteuning van de eindopdracht waar zelfstandig een casus uitgewerkt moet worden.

Bij de opgaven en opdrachten gebruikt u de tool Objectiver om modellen te maken. Zie ook bij de leermiddelen.

Het kan zijn dat er wel eens verwezen wordt naar stof in de niet behandelde hoofdstukken. Dan graag even terugbladeren.

U gaat diep in op één methode en op de bijbehorende technieken. De stof die behandeld wordt in deze leereenheden, dient u aan het eind van de cursus echt te beheersen; de nadruk ligt dus op kunnen.

Ten behoeve van het tentamen werkt u voor het onderdeel Requirements Engineering (leereenheden 6 tot en met 11) zelfstandig aan een casus met de bedoeling om zelfstandig de gewenste modellen uit te werken. Op de uitwerking wordt een keer feedback gegeven. Zie ook de informatie hieronder bij de paragraaf *Tentaminering*.

De verschillende opdrachten kunt u terugvinden op Studienet of opvragen bij de docent.

2 Leerdoelen

Na het bestuderen van de cursus wordt verwacht dat u kennis en inzicht heeft verworven over

- algemene principes die een rol spelen bij softwarelifecycles
- een aantal procesmodellen voor softwareontwikkeling
- een aantal methoden voor procesverbetering
- factoren die van belang zijn voor goed softwaremanagement
- aspecten die een rol spelen voor de kwaliteit van software.

Daarnaast wordt van u verwacht dat u

- de verworven inzichten kunt toepassen op een concrete situatie en daarvan verslag kunt doen
- de rol van requirements engineering in de lifecycle van een systeem kent en inzicht heeft in de belangrijkste obstakels in dit proces
- de verschillende technieken kent voor risicobeheersing en weet wanneer welke techniek geëigend is
- de meest gebruikte technieken kent voor de specificatie van requirements en inzicht heeft in de sterke en zwakke punten van deze technieken
- weet wat binnen de KAOS-methode bedoeld wordt met goals, objects, agents en operations, welke eigenschappen deze hebben en in welke typen ze zijn onder te verdelen

- in staat bent om voor een te ontwikkelen systeem, gebruikmakend van de tool Objectiver, de volgende modellen te construeren:
 - een goal model ter representatie van wat met behulp van het systeem bereikt moet worden
 - een obstacle model ten behoeve van risicoanalyse en als uitbreiding op het goal model
 - kunt uitleggen hoe u bij het opstellen van een goal en obstacle model gebruik kunt maken van scenario's.
- op grond van een casus zelf een goal model kunt opstellen met mogelijk bijbehorende obstacles. Dit model wordt beoordeeld en is onderdeel van het tentamen.

3 Voorkennis

Er wordt uitgegaan van het kennisniveau van een afgeronde bacheloropleiding Informatica.

4 Cursusmateriaal

Het cursusmateriaal bestaat uit:

- een werkboek: leidraad voor de cursus met leereenheden, een aantal artikelen, opgaven en aanvullende informatie bij de artikelen (onder meer een korte beschrijving in het Nederlands)
- het tekstboek 'Requirements Engineering, From System goals to UML models to Software specifications' van Axel van Lamsweerde, John Wiley, 2009.
- een cursussite op Studienet met actuele informatie over begeleiding en tentamen, aanvullingen en errata, artikelen, weblinks en de ingang tot de discussiegroep.

De cursus maakt gebruik van de KAOS-tool Objectiver. Dit is een professionele tool die u uitsluitend mag gebruiken voor het bestuderen van de cursus. Om de tool na installatie te kunnen gebruiken, hebt u een sleutel nodig die u kunt aanvragen bij de examinator. De geldigheidsduur van die sleutel is veertien maanden; u moet de sleutel daarom pas aanvragen als u daadwerkelijk begint met het bestuderen van de cursus. Details over de installatie van de tool en het verkrijgen van de sleutel vindt u op Studienet.

5 Studeeraanwijzingen

Een onderdeel van dit werkboek vormen de opgaven en oefenopdracht. Deze zijn toegevoegd als hulpmiddel om zowel het leesproces te levendigen als toe te spitsen op de essentie. Het uitwerken van opgaven en oefenopdracht vormt een verplichting voor deelname aan het tentamen. U stuurt de antwoorden aan de begeleider en krijgt van deze feedback. Een en ander wordt ook besproken in de online sessies. Dit kan individueel zijn of groepsgewijs tijdens de begeleidingsbijeenkomsten. Precieze gegevens over inleveren van de opgaven en opdrachten vindt u op Studienet, met een selectie van de opgaven waarvan verwacht wordt dat u deze uitwerkt.



6 Begeleiding en tentaminering

Begeleiding

Bij deze cursus wordt een begeleide cyclus aangeboden met een aantal online begeleidingsbijeenkomsten. Hier worden opgaven besproken en toelichting gegeven op de leerstof, inclusief uitleg bij de eindopdrachten. De begeleidingscyclus heeft een duur van tien weken en een aansluitende tentamenweek.

Een volledig studieplan kunt u vinden op Studienet met per week aangegeven wat verwacht wordt dat u bestudeert en hoeveel tijd dit gaat kosten.

Voor verdere informatie kunt u de cursussite op Studienet raadplegen.

Tentaminering

De vorm van het tentamen is mondeling. Vertrekpunt daarbij is het eerdergenoemde korte verslag en de uitwerking van de opdracht voor Requirements Engineering, die ten minste een week voor het tentamen ingeleverd moeten zijn. De opdrachten zijn te vinden op Studienet of te verkrijgen via de docent.

Daarnaast zullen tijdens de tentamensessie ook andere onderwerpen uit de cursus besproken worden. Het tentamen kan dus over de gehele leerstof gaan. Voorwaarde voor deelname aan het tentamen is dat tijdig aan alle verplichtingen, opdrachten en verslag, is voldaan. Duur van het tentamen is 75 minuten.

Het cijfer voor de cursus bestaat uit het gewogen gemiddelde van de mondelinge bevraging over de gehele stof (gewicht 0,7) en de opdracht Requirements Engineering (gewicht 0,3).

Voor tentamendata kunt u de cursussite op Studienet raadplegen.

ⁱ zie daarvoor bijvoorbeeld T. Bhuvanewari et al, A Survey on Software Development Life Cycle Models, in *International Journal of Computer Science and Mobile Computing* Vol.2 Issue. 5, May- 2013, pg. 262-267, 2013.

ⁱⁱ K. Kumar and S. Kumar, A rule-based recommendation system for selection of software development life cycle models; In *Proceedings of ACM SIGSOFT Software Engineering Notes*. 2013, 1-6.