

Leidraad
Methodeselectie
voor softwareontwikkeling



Inhoudsopgave

1.	Inleiding	2
2.	Het maken van een plan van aanpak	3
3.	Het proces	5
	3.1. Het spectrum van mogelijkheden	5
	3.2. Kiezen: formeel of informeel?	6
	3.3. Kiezen: lineair of iteratief?	7
	3.4. Andere factoren bij het kiezen van een methode	8
4.	De technieken.....	9

1. Inleiding

Dit document is bestemd voor alle medewerkers die te maken hebben met analyse en ontwerp binnen Ordina SI&D. Het biedt een handvat om een werkproces te selecteren voor het uitvoeren van analyse- en ontwerpopdrachten.

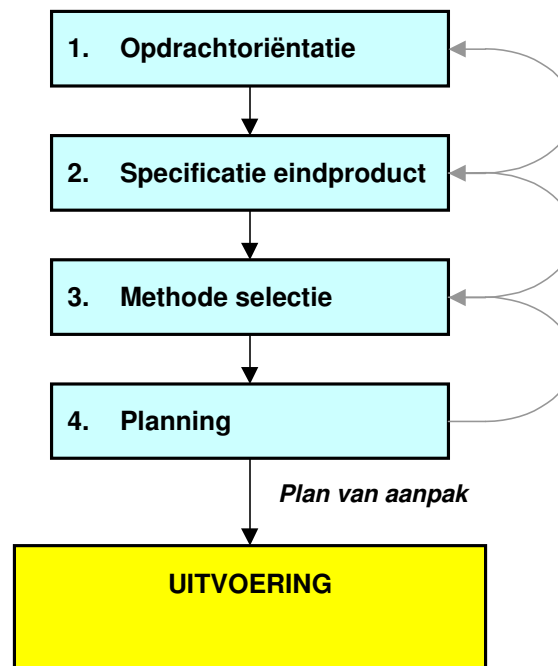
Hans Admiraal

www.admiraal.ws
admiraal@aol.nl
06 250 948 22

2. Het maken van een plan van aanpak

Aan het begin van elk IT-project of -deelproject wordt er een plan van aanpak gemaakt waarin wordt aangegeven wat de opdracht is en hoe die wordt aangepakt, in termen van tijd, geld, kwaliteit, informatie en organisatie.. De keuze van een ontwikkelmethodiek is een onderdeel hiervan, want deze heeft direct betrekking op bovengenoemde projectpijlers. In deze paragraaf plaatsen wij het selecteren van een methode in die context.

Het maken van een plan geschiedt in vier stappen, die zijn weergegeven in de onderstaande figuur.



Figuur 1. Vier stappen voor het bepalen van de aanpak.

De stappen zijn:

1. **Opdrachtoriëntatie.** Allerlei aspecten van de opdracht worden in kaart gebracht, zowel de doelstellingen als de omgeving en de kaders waarbinnen de opdracht zal worden uitgevoerd.
2. Het **specificeren** van de eisen ten aanzien van het **eindproduct**. De eisen richten zich op het doel van het product en niet op de methode of de technieken. Er worden waar mogelijk meetbare criteria geformuleerd, zodat een "moving target" wordt vermeden.
3. Het **selecteren van een methode** en het toesnijden van die methode op de concrete situatie, bijvoorbeeld door elementen toe te voegen en/of weg te laten. De methode zoals die in het plan van aanpak beschreven zal worden, bestaat uit de volgende elementen:
 - a. Het **proces**. Dit beschrijft de levenscyclus van het project. Het proces geeft aan wie wanneer wat doet. Het definieert de rollen en hun verantwoordelijkheden, activiteiten en mijlpalen.
 - b. De gebruikte **technieken**. Technieken geven een nadere invulling van de producten. Op het gebied van analyse en ontwerp wordt meestal gebruik gemaakt van tekst, schema's en diverse soorten prototypes.
 - c. **Hulpmiddelen**. Te denken valt aan softwarehulpmiddelen zoals een CASE tool en een tekstverwerker, maar ook aan computer hardware en kantoorartikelen.
4. Het maken van een **planning** waarin de diverse elementen van de methode worden uitgezet tegen de tijd, mensen en middelen.

De stappen worden bij voorkeur niet strikt opeenvolgend doorlopen, maar kunnen elkaar beïnvloeden. Zo kan de keuze voor een methode vragen oproepen ten aanzien van de opdracht of de eisen, of kan de planning er toe leiden dat de scope van de opdracht wordt ingeperkt.

Deelplannen

Let er op dat er binnen een opdracht vaak deelopdrachten bestaan die elk een eigen plan van aanpak vereisen. Ook als je wordt ingezet voor een project waarvoor er al een plan van aanpak is gemaakt, is het verstandig om voor jouw concrete deelopdracht opnieuw een plan van aanpak te schrijven, eventueel in afgeslankte vorm; jouw opdracht, jouw opdrachtgever, jouw eindproduct etc. kunnen immers totaal anders zijn dan die van het project als geheel. Misschien kunnen de diverse deelplannen worden opgenomen in het overkoepelend plan van aanpak.

Advies inwinnen

Twijfel je over de keuze van een methode? Deze leidraad kan inzicht verschaffen, maar vraag ook altijd advies aan een collega met een brede methodische ervaring.

Accorderen

Laat de opdrachtgever het plan accorderen, zodat je hem/haar naderhand kunt wijzen op de gemaakte afspraken. Daarnaast is het goed de verwachtingen van de opdrachtgever te managen.

3. Het proces

In deze paragraaf wordt nader ingegaan op stap 3.a van het schrijven van een plan van aanpak, namelijk het bepalen van het proces, als onderdeel van de te volgen methode.

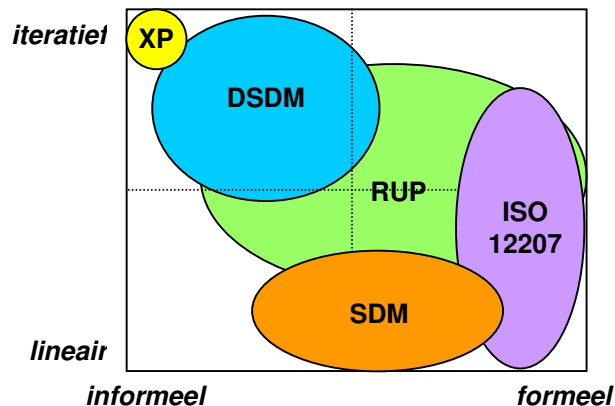
3.1. Het spectrum van mogelijkheden

We beschouwen twee eigenschappen van processen, die een spectrum van mogelijkheden opleveren:

- Hoe formeel is het proces? Een proces met weinig formaliteiten, weinig documentatie en veel prototyping is geschikt voor een klein team dat veel moet experimenteren. Een strak geregeld proces met gedetailleerde documentatie is geschikt in een omgeving waarin veel betrokkenen verschillende belangen hebben en er een complex systeem gebouwd moet worden onder redelijk stabiele eisen.
- Is het proces lineair of iteratief? Bij een lineair proces doorloopt men top-down een vast aantal fasen, bijvoorbeeld analyse, ontwerp, bouw, test. Dit is overzichtelijk voor iedereen en eenvoudig te plannen. Bij een iteratief proces doorloopt men deze fasen een aantal malen, waarbij na elke iteratie een nieuwe versie van het eindproduct wordt opgeleverd. Tussenvormen zijn ook mogelijk, bijvoorbeeld een deel lineair en een deel iteratief doen, of lang durende iteraties uitvoeren met elk een lineair deelproces.

Ook al wordt er gekozen voor een standaard methode, toch zal er altijd een afweging gemaakt moeten worden in beide dimensies: hoe formeel passen we de methode toe en waar en hoe gaan we itereren?

In de volgende figuur is een aantal bekende methoden gepositioneerd in het spectrum.



Figuur 2. Positionering van methoden in twee dimensies van proceskenmerken.

De figuur geeft aan, dat elke methode een zekere interpretatievrijheid biedt om meer of minder formeel en meer of minder iteratief te werk te gaan. Alleen Extreme Programming (XP) is erg radicaal. RUP daarentegen bestrijkt een groot deel van het spectrum, waardoor er veel aandacht besteed moet worden aan het toesnijden van deze methode op een concrete opdracht. In RUP-terminologie heet dit het maken van een 'development case'. SDM en DSDM benoemen dit niet zo, maar voor die methoden geldt in mindere mate hetzelfde. DSDM staat in het spectrum relatief informeel gepositioneerd, maar het is mogelijk om extra formaliteiten toe te voegen, bijvoorbeeld met PRINCE 2, om ze zo formeel te maken als men wenst. Het is dus niet voldoende om een bepaalde standaard methode te kiezen, maar men moet ook bepalen welke elementen van die methode gebruikt kunnen worden, welke formaliteiten er zijn (bijv. accordering van tussenproducten of een change management procedure) en welke iteraties er eventueel worden toegepast.

Positionering van SMART

SMART is oorspronkelijk ontwikkeld binnen Ordina Finance, maar wordt nu onderhouden door een concernbreed Competence Center. SMART is gebaseerd op DSDM en kent dezelfde positionering in het spectrum. In aanvulling op DSDM geeft SMART aan welke UML-technieken je kunt gebruiken.

Positionering van SLIM

SLIM is ontwikkeld en wordt gebruikt binnen Ordina TTI / Technical Automation. SLIM is een pragmatische invulling van de ISO-12207 (J-STD-016) standaard.

Positionering van PRINCE 2

De eerdergenoemde methode PRINCE 2 wordt in dit document niet nader besproken, aangezien het een algemene projectmanagementmethode is en niet een methode voor analyse en ontwerp.

Positionering van UML

UML is niet gepositioneerd in het spectrum van methoden, aangezien het slechts een verzameling technieken is en het geen proces beschrijft.

Positionering van Agile Manifesto

The Agile Manifesto is geen methode, maar een visie, waarin een sterk informele en iteratieve aanpak wordt gepropageerd.

3.2. Kiezen: formeel of informeel?

Een opdracht die binnen een week moet zijn afgerond, kan misschien volledig informeel uitgevoerd worden, maar in alle andere gevallen zijn bepaalde formaliteiten geboden. In de eerste plaats is dat het schrijven van een plan van aanpak. De volgende tabel geeft een aantal overwegingen die kunnen helpen bij het bepalen van de mate van formaliteit in de aanpak.

	Informeel	Formeel
Doorlooptijd	kort	lang
Complexiteit inhoudelijk	eenvoudig	complex
Complexiteit projectorganisatie	eenvoudig	complex
Teamgrootte	klein	groot
Stabiliteit van de doelstellingen	instabiel	stabiel
Onderling vertrouwen	groot	klein
Cultuur van het team / de organisatie	informeel	formeel

De cultuur van het team is nauw verbonden met de karaktereigenschappen van de medewerkers. Sommige mensen werken liever informeel, anderen liever formeel. Onderken je eigen natuurlijke neiging en beheers die. Vraag naar de voorkeur van de opdrachtgever en van de medewerkers en houd daar rekening mee.

Er zijn verschillende soorten van formaliteiten die gehanteerd kunnen worden. In de onderstaande lijst wordt elk aspect gevolgd door enkele vragen die bij het bepalen van de aanpak beantwoord moet worden.

- Accordering van tussenproducten door opdrachtgever en eventueel andere betrokkenen.
Hoe formeel? Door wie? Hoe vaak?
- Planning.
Hoe gedetailleerd? Hoe formeel komen de tijdschattingen tot stand?
- Overlegstructuur.
Welke soorten overleg zijn er en in welke frequentie? Zijn er agenda's, notulen en besluitenlijsten?
- Voortgangsbewaking en prestatiemeting.
Hoe gedetailleerd wordt de tijdsbesteding verantwoord? Hoe wordt het resultaat gemeten, in kwantiteit en in kwaliteit? Is er een lijst van actiepunten?
- Change management en versiebeheer.

Moeten veranderingen worden geaccordeerd? Worden ze gelogd? Wordt er een tool gebruikt voor versie- en configuratiemanagement? Is er een systeem waarmee foutrapportages en wijzigingsvoorstellen formeel kunnen worden ingediend?

- Tracking & tracing.
Worden de relaties tussen doelstellingen en analyse- en ontwerponderdelen vastgelegd, zodat men bij een verandering van een doelstelling kan nagaan wat de impact op het ontwerp is (tracking) en zodat men kan nagaan waarom een bepaald deel van het ontwerp ontstaan is (tracing)?
- Rollen en verantwoordelijkheden.
Hoe gedetailleerd worden de rollen en verantwoordelijkheden binnen het project vastgelegd en toegekend aan medewerkers?
- Evalueren, toetsen, testen.
In hoeverre wordt het V-model toegepast? Wordt de analyse formeel getoetst aan de doelstellingen? En het ontwerp aan de analyse? Kan de analyse en/of het ontwerp dienen om de uiteindelijke software ondubbelzinnig te testen?
- Analyse- en ontwerptechnieken.
Ligt de nadruk op vrije tekst of op formele schema's? Hoe strikt zijn de symbolen en begrippen gedefinieerd en hoe strikt wordt er aan die definities vastgehouden?

3.3. Kiezen: lineair of iteratief?

In de IT-wereld wordt er tegenwoordig meer vertrouwen gesteld in een iteratieve aanpak dan in een lineaire aanpak, o.a. omdat een iteratieve aanpak beter bestand is tegen wijzigingen in doelstellingen en eisen ("voortschrijdend inzicht"), gebruikersbehoeften en technologische mogelijkheden.

Bij de lineaire aanpak wordt na elke fase een tussenproduct opgeleverd en pas na de laatste fase het eindproduct. Bij een iteratieve methode wordt na elke iteratie een deel van het eindproduct opgeleverd. De eerste iteratie richt zich op de meest bedrijfskritische en risicovolle delen en latere iteraties richten zich hoe langer hoe meer op eenvoudige en minder urgente componenten. Om in iteraties te kunnen werken, is het dus belangrijk dat het eindproduct kan worden verdeeld in goed gedefinieerde stukken, bijvoorbeeld door het toekennen van verschillende prioriteiten aan de eisen die er aan gesteld worden.

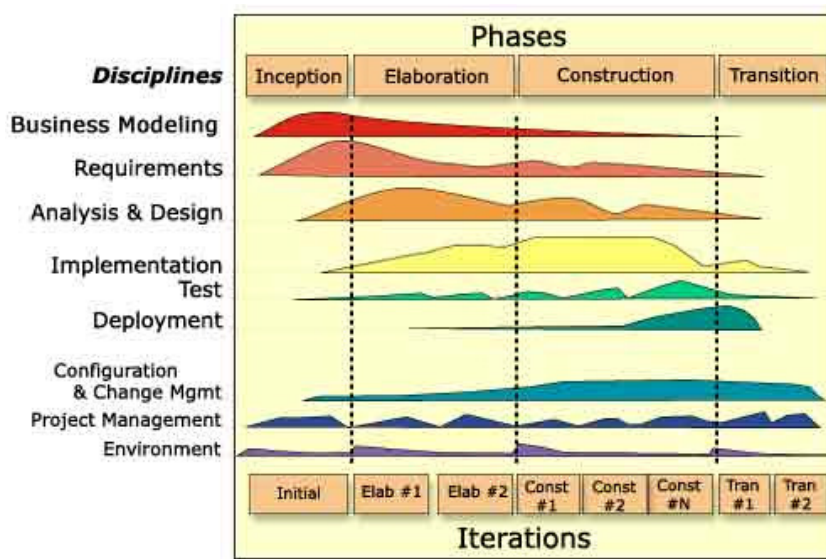
Bij een extreem lineaire aanpak wordt de teamsamenstelling aangepast aan de fase: eerst komen bijvoorbeeld de analisten, daarna de ontwerpers en daarna de bouwers. Bij een extreem iteratieve aanpak moeten al deze disciplines continu in het team aanwezig zijn. Overigens is het ook bij een lineaire aanpak altijd belangrijk om al tijdens het maken van het functioneel ontwerp ook de technische specialisten, de testers, key-users uit de gebruikersorganisatie en de applicatiebeheerders er bij te betrekken.

De bovenstaande en andere overwegingen zijn samengevat in de volgende tabel.

	Lineair	Iteratief
Eindproduct is deelbaar / eisen zijn geprioriteerd	nee	ja
Samenstelling van het team	gespecialiseerd	multidisciplinair
Aard van de opdracht	routine	experimenteel
Ervaring met iteratieve aanpak	weinig	veel
Vertrouwen in iteratieve aanpak	weinig	veel
Haalbaarheid van regelmatige introductie van een nieuwe versie	kostbaar	eenvoudig
Het eisenpakket is gesteld in termen van:	technische oplossingen	bedrijfsdoelstellingen

De keuze tussen lineair en iteratief is niet binair. Er zijn verschillende manieren om een beetje lineair en/of een beetje iteratief te werken. De onderstaande lijst geeft een aantal opties, om het vinden van de juiste balans te vergemakkelijken.

- Bij een lineaire aanpak is het vaak verstandig een flinke overlap tussen de fasen aan te brengen. Tijdens een overlapperperiode kan men informeel itereren;
- Bij een iteratieve aanpak is het vaak verstandig om de eerste fasen, zoals haalbaarheidsstudie en bedrijfsstudie, lineair te doen en daarna pas te itereren. Dit is wat DSDM en SMART voorstaan;
- Het is zelfs mogelijk om alleen tijdens de bouw- en testfase te itereren en de rest lineair te doen, of juist andersom;
- Een iteratief proces kan opgedeeld worden in een vast aantal fasen, waarbij binnen elke fase wordt geïtereerd. Dit is wat RUP voorstaat (zie Figuur 3).
- Het aantal iteraties kan worden beperkt tot twee, of uitgebreid tot welk aantal men ook maar wil. Evenzo kan de duur van een iteratie variëren van een week tot een jaar.



Figuur 3. Het RUP proces (iteratief)

3.4. Andere factoren bij het kiezen van een methode

Behalve dat er een positie gekozen moet worden binnen de dimensies van formeel/informeel en lineair/iteratief, zijn er nog een aantal andere factoren waarmee men rekening moet houden bij het selecteren en toesnijden van een methode.

- **Aansluiten bij wat er al is**

Een compleet IT-project begint met een probleemstelling van de klant en eindigt met een IT-oplossing. Een compleet IT-proces dekt dit hele traject. Een opdracht omvat echter vaak niet een compleet IT-project, maar slechts een deel ervan, bijvoorbeeld het maken van een ontwerp op basis van een gegeven bestek. In dat geval is het belangrijk om na te gaan volgens welke methode het overkoepelende project wordt gehanteerd en om daarvoor het uit te voeren deelproject bij aan te sluiten. In deze leidraad gaan we ervan uit dat er door de opdrachtgever nog geen methode is gekozen.

- **Aansluiten bij kennis en ervaring**

De kennis en de ervaring van de teamleden speelt een grote rol in het kiezen van een methode. Uiteraard kan het verstandig zijn iets nieuws te introduceren, maar doe het altijd met mate.

- **Participatie van gebruikers**

Als in het team voldoende gebruikers vertegenwoordigd zijn en als het team het mandaat heeft gekregen om het systeem vorm te geven, dan kan men met weinig formaliteiten en veel prototyping een uitstekend resultaat behalen. DSDM noemt dit als voorwaarde om die methode te gebruiken.

4. De technieken

Deze leidraad gaat niet uitgebreid in op de technieken die gebruikt kunnen worden, maar geeft wel enige aanwijzingen voor het selecteren van technieken.

4.1.1. Algemeen

- Indien de gekozen methode voldoende invulling geeft aan het gebruik van technieken, dan is het raadzaam om daarnaar te handelen. Dit is bijvoorbeeld het geval bij SMART;
- Sommige methoden bieden veel vrijheid ten aanzien van de te gebruiken technieken. In dat geval moet er in het plan van aanpak aangegeven worden welke technieken er gebruikt gaan worden. SDM en DSDM zijn bijvoorbeeld zowel geschikt voor een gestructureerde als een objectgeoriënteerde benadering (meer hierover in paragraaf 4.1.4);
- Gebruik bij het tekenen van schema's zoveel mogelijk UML.

4.1.2. Teksten en schema's

- Tekst en schema's vormen een belangrijke combinatie, maar welke van de twee is maatgevend? Kies expliciet tussen ofwel primair tekst, met schema's ter illustratie, ofwel primair schema's met tekst als aanvulling;
- Alle schema's moeten, voorzien van een toelichting, begrijpelijk zijn voor mensen zonder kennis van IT. Zeker bij informatieanalyse en functioneel ontwerp is communicatie met materie-deskundigen en gebruikers immers essentieel.

4.1.3. Prototype

Wanneer je het maken van een prototype opneemt in het plan van aanpak, maak dan duidelijk om welk soort prototype het gaat:

- een user interface prototype (dit is vaak een goed middel om de functionaliteit met gebruikers te bespreken);
- een technologische proof-of-concept.

Daarnaast moet ook een expliciete keus gemaakt worden voor één van de volgende twee opties:

- Een gedeeltelijk werkend programma dat bedoeld is om uit te groeien tot eindproduct;
- Een prototype dat bedoeld is om uiteindelijk weggegooid te worden. Bijvoorbeeld:
 - een verzameling schetsen op papier;
 - een verzameling afbeeldingen, gemaakt in Visio of Powerpoint;
 - een statische web site;
 - een gedeeltelijk werkend programma.

4.1.4. Paradigma

Het is verstandig om een consistent paradigma te kiezen voor alle technieken. Een aantal opties wordt hieronder opgesomd (referenties naar literatuur moeten nog toegevoegd worden):

- Gestructureerde aanpak (SA/SD). Deze kenmerkt zich door een strikte scheiding tussen het datamodel en de functionaliteit van het systeem. Voor de functionaliteit voert men een top-down decompositie uit (modules, submodules, functies). De modules en functies zijn black boxes, maar het datamodel is een white box. Kies uitsluitend voor deze aanpak als het team een modernere techniek niet aankan;
- Objectoriëntatie (OO). Hierbij worden de gegevens ingekapseld door bijbehorende functies, waardoor ook de gegevens in black boxes ("objecten") worden opgenomen. Verder spelen begrippen als instantiatie en overerving een rol;
- Component-based development (CBD). Een component is een speciaal soort object. Qua grootte lijken componenten echter meer op modules: kleine, basale objecten zien we in de applicatie niet terug als afzonderlijke componenten. Componenten onderscheiden zich door hun onderlinge onafhankelijkheid. Terwijl modules vaak gekoppeld zijn door een gemeenschappelijk datamodel en objecten gekoppeld zijn door o.a. overerving, werken componenten alleen met elkaar samen via hun interfaces;

- Service-oriënted architecture (SOA). Services zijn componenten die aangeboden worden via het internet (web services) of een ander netwerk, om gebruikt te worden binnen diverse applicaties, bijvoorbeeld applicaties van derden. De service-software wordt daarbij niet verkocht of gekopieerd, maar de service blijft draaien op de locatie van de aanbieder.

4.1.5. Andere technieken

Bij “technieken” denken velen meteen aan schema- technieken. Er zijn echter vele andere soorten technieken. Prototyping is al genoemd. Hieronder volgt een greep uit het overige aanbod:

- Timeboxing;
- use cases;
- facilitated workshops;
- CRC cards;
- analysis patterns;
- design patterns.