
Reader 5

ERP implementatie, verandering en rapportage

Generic software: een standaard softwarepakket die is ontworpen door de leverancier.

Implemented packet: een geïmplementeerd (standaard) softwarepakket dat is geconfigureerd, uitgebreid of is aangepast.

Organizational information system: een organisatie die een informatiesysteem toepast in de vorm van een (standaard) ERP-systeem.

Bij een generieke software bestaat het artefact alleen uit software. De software heeft ook levenscyclus: defined, designed, built, tested en released (of maintained), frozen en vervolgens withdrawn. Kleine verbeteringen aan de software kunnen worden gedaan door middel van service package. Daarnaast kunnen grote verbeteringen aan de software worden gedaan door middel van releases. Een upgrade van softwarepakket S_n leidt tot het ontstaan van softwarepakket S_{n+1} . Het nieuwe softwarepakket is een nieuw artefact met een eigen levenscyclus.

Als een pakket wordt geïmplementeerd voor een klant, leidt dit tot een nieuw artefact dat uit software en data bestaat.

Daarnaast is een organisatie die een informatiesysteem gebruikt geen artefact, maar bestaat het systeem uit mensen, software, middelen en data. Dit tezamen informeert de organisatie en haar stakeholders.

In deze syllabus heeft een ERP-systeem betrekking op een geïmplementeerd pakket.

Een voordeel van een ERP-systeem is de toegankelijk tot data voor het management. Daarbij is het wel lastig om deze informatie om te zetten in bruikbare informatie.

Configuratie verwijst naar het bepalen van de parameters van een ERP-pakket op basis van de keuzes van een organisatie. Hierdoor kunnen bepaalde functionaliteiten in- of worden uitgeschakeld en kunnen bepaalde softwaremodules compleet worden gemaakt. Daarnaast bevat configuratie ook de parameters van de samenstelling van verschillende modules of softwarecomponenten. Maar ook bevat het specificatie en het personaliseren van het ERP-pakket. Tot slot kan met behulp van configuratie reporten worden ontworpen met de software van de leverancier. In de onderstaande tabel zijn de opties voor configuratie opgenomen.

Parameter setting	Er kan worden gekozen welke uitvoeringen, processen en functies moeten worden opgenomen in de software. Daarnaast kunnen bepaalde functies in en uit worden geschakeld. Er kunnen veel parameters zijn, waardoor organisaties beslissingsbomen, tabellen, modellen, etc. gebruiken.
Screen configuration and screen masks	Het gebruikersscherm wordt vaak ingesteld aan de voorkeuren van de gebruiker. Leveranciers leveren hiervoor opties om deze schermen te configureren. Screen masks worden gebruikt voor het creëren van een nieuwe sessie door bestaande sessies samen te voegen. Dit is handig voor als veel dingen in het systeem niet worden gebruikt tijdens de implementatie. Hierdoor worden sessies vereenvoudigd en wordt het gebruiksgemak vergroot.
Configuration standard ERP reports	Door middel van parameters kan worden bepaald hoe de standaard reporten eruit komen te zien. Bijvoorbeeld als een report op papier of op een scherm getoond moet worden, kan dit door middel van parameter worden aangepast.

Tabel 1: Opties voor configuratie

Door middel van configuratie transformeert een generieke ERP-software in een geïmplementeerde ERP-software. De verschillende configuraties die kunnen worden gedaan zijn groot. In principe is een ERP-softwarepakket altijd in een configurerende staat. Als er iets verandert in het ERP-systeem door configuratie moet dit gedocumenteerd worden, waardoor het nodig is dat medewerkers hiervoor getraind worden en testen soms noodzakelijk is. Ook is het mogelijk om een geïmplementeerd ERP-pakket van een configuratie in een andere configuratie te transformeren. Dit is een voorbeeld van een ERP-verandering. Customization daarentegen verwijst naar het veranderen of nieuwe functionaliteiten aanbrengeen aan de software.

De term modification verwijst naar veranderingen in de ERP source code. Hieronder valt volgens sommige literatuur de connectie die gelegd wordt tussen verschillende systemen. Het leggen van deze connectie kan niet worden opgevat als een verandering, maar als een ERP-systeem dat onderdeel uitmaakt van een groter informatiesysteem. Dit brengt geen verandering aan in het ERP-systeem, alleen hoeft er een extra interface worden gecreëerd. Hiervoor wordt de term extension gebruikt. In tabel 2 op bladzijde 5 van de vijfde reader zijn situaties weergegeven waarbij er extension en modification plaatsvindt. Als er een extensie plaatsvindt moet dit gedocumenteerd, getest en getraind worden. Als er een upgrade plaatsvindt is het van belang dat er wordt gecontroleerd. Hierdoor zal er weer opnieuw gedocumenteerd worden, getest en worden getraind. Als een extensie gebeurt naar een third party software heeft het bedrijf software interfaces buiten de onderneming, hierdoor vergroot het beveiligingsrisico.

Als er geen modificatie plaatsvindt, dan wordt dit vanilla genoemd. Als er wel een modificatie wordt aangebracht kan dit als doel hebben het vervangen van de bestaande functionaliteit van het standaardsysteem door nieuwe functionaliteiten. Hierdoor verandert het datamodel en de parameters.

Er kunnen zowel kleine als grote modificaties plaatsvinden. Door modificaties toe te passen kan het succes van de implementatie worden bepaald.

Voor het creëren van meerdere interfaces is de programmering afhankelijk van de versies die voor de systemen gebruikt worden. Daarnaast is het systeem erg gevoelig voor upgrades van beide kanten. Als gevolg moeten de interfaces gecontroleerd worden. Nieuwe technologieën zoals webservices kunnen deels het probleem wegnemen.

De scope van een ERP-systeem betreft welke onderdelen van een organisatie ondersteund worden. Dit kan bijvoorbeeld variëren van de gehele organisatie of delen hiervan. Door middel van de scope kan worden vastgesteld welke functies van het ERP-systeem wordt ingeschakeld dan wel uitgeschakeld. De

scope van een ERP-systeem kan single-site zijn. Dit is een fysieke locatie van bijvoorbeeld een fabriek een magazijn of een winkel. Echter ontstaan de meeste problemen in een organisatie waarbij er sprake is van meerdere locaties met een ERP-systeem waar tussen transacties plaatsvinden. In dit geval is er sprake van een multi-site. Multi-sites solutions kunnen worden toegepast als er sprake is van zowel decentralisatie als centralisatie van de bedrijfsvoering.

Multi-site implementation moet niet verward worden met multi-single site solution. Bij multi-single site solution heeft elke locatie zijn eigen data en configuratie, zelfs al is de configuratie gestandaardiseerd tussen locaties en het hoofdkantoor. Daarentegen bij muti-site implementaties worden niet alleen de locatie geconfigureerd, maar ook de bedrijfsrelaties tussen verschillende locaties en het hoofdkantoor. Als gevolg neemt de complexiteit toe. De scope is een variabele en kan twee waardes aannemen: single site en multi-site.

Voor het gebruik maken van een het ERP-systeem (ERP application deployment) wordt er steeds meer gebruik gemaakt van cloud computing. Als het ERP-software wordt geïnstalleerd op een eigen computer wordt dit on premise genoemd. On premise is verder uitgewerkt in twee andere modellen: Application Service Provision (ASP) en Software as a Service (SaaS). Het voordeel van ASP is dat verschillende klanten gebruik maken van dezelfde release en servicepakketten van de standaard software. Als gevolg worden de toepassing, uitbreiding en upgraden vergemakkelijkt. Elke klant heeft daarbij wel zijn eigen software geïnstalleerd bij de leverancier. De waarde van de ASP zit in het delen van de resources onder verschillende klanten. Het kent wel als nadeel dat er beperkingen zijn in het toepassen van veranderingen, omdat upgrades tezamen met andere klanten worden gedaan.

Daarentegen in het SaaS-model gebruikt elke klanten dezelfde installatiecode. Als gevolg is er geen mogelijkheid om een andere code te gebruiken. Daarnaast is het wel mogelijk om eventueel uit te breiden naar andere softwarepakketten. SaaS biedt hierdoor de mogelijkheid om per individuele klant aanpassingen te maken door middel van uitbreidingen. Aanpassingen kunnen dus niet aangebracht in de software.

Standaard softwarepakketten kunnen worden geïdentificeerd aan de hand van het release identification number en een service pack nummer, welke correspondeert met kleinere en grotere updates. Minor upgrades zijn kleine aanpassingen in de software waarbij nieuwe functionaliteit wordt toegebracht. Door minor upgrades veranderen de parameters en de datastructuur niet. Hierdoor is het alleen noodzakelijk dat na een minor upgrade de software wordt gecontroleerd en getest wordt door de consument. Een service pack haalt kleine fouten uit de software, waardoor de impact klein is. Het wordt ook wel een Legal Change Patch (LCP) genoemd. Service packs zijn kleinere verbeteringen in de software, waardoor een service pack onder een minor upgrade valt.

Een major upgrade daarentegen maakt wel veranderingen in de functionaliteit van het systeem. Als gevolg veranderen de parameters, de interactiepatronen en de datastructuur wel degelijk. Hierdoor moet er vaak een her implementatie plaatsvinden: het ERP-pakket moet opnieuw geconfigureerd worden, getest, gedocumenteerd, werknemers opnieuw getraind worden en de ERP-databases zal opnieuw gemigreerd worden. Eventueel moeten uitbreidingen opnieuw getest worden, herontworpen worden en de aanpassingen moeten beoordeeld worden en opnieuw worden ingesteld. Als de functionaliteit na de upgrade goed is verlopen, hoeven eventueel de uitbreidingen niet aangepast worden.

Als er een (nieuwe) configuratie wordt toepast, is het van belang dat elke configuratie goed getest wordt. De impact van configuratie op de technologie doet zich voor in het testen. Door de kwaliteit van de configuratie heeft een grote invloed op de prestatie van het proces en de bedrijfsuitkomst. Door middel van configuratie kan een onderneming voordelen behalen. Als er een configuratie plaatsvindt is het van

belang dat dit wordt gedocumenteerd en dat de gegevens worden gebruikt in trainingen voor de gebruikers. Het opnieuw configureren heeft geen invloed op de de beveiligingsrisico's en de kosten voor het upgraden. Configuratie biedt de onderneming een verbetering van het bedrijfsproces.

Ook de impact van (nieuwe) extensies op technologie uit zich in de vorm van testen. Door het toepassen van een extensie ontstaat er een beveiligingsrisico. Bovendien is het van belang dat extensies worden gedocumenteerd en dat er een training volgt voor de gebruikers. Als er een upgrade plaatsvindt, is het van belang dat er opnieuw wordt geïnspecteerd en zo nodig opnieuw wordt getest en herontwikkeld. Als de extensie juist wordt uitgevoerd kan dit een competitief voordeel opleveren voor de onderneming.

ERP-systemen maken een deel uit van een groter informatiesysteem. Het is van belang dat het informatiesysteem een compleet en consistente applicatie heeft voor haar gebruikers. Door de combinatie van ERP en andere applicaties kan het voorkomen dat er gaten en dubbele informatie in het informatiesysteem aanwezig zijn. Om dit te vermijden worden aanpassingen gedaan aan het ERP-systeem en andere applicaties. Als gevolg zal opnieuw een configuratie worden gedaan of zal er een wijziging optreden.

Als ERP-systemen en externe locaties met elkaar in verbinding staan, kan dit een beveiligingsrisico met zich meebrengen. Als ERP intern met elkaar in verbinding staan kan door middel van een firewall de beveiliging gegarandeerd worden. Het is dus van belang dat het risico aspect wordt overdacht wanneer externe locaties met elkaar informatie uitwisselen. Elke nieuwe interface brengt beveiligingsrisico's met zich mee.

De impact van modificatie op technologie is dat naast het testen ook het risico wordt gelopen dat de leverancier de onderneming niet meer steunt. Hierbij speelt de grote van de modificatie ook een rol. Door een wijziging aan te brengen in de code leidt dit tot andere contracten dan wanneer er geen wijzigingen worden aangebracht. Door de modificatie stijgen de kosten voor de onderneming, maar daarnaast wordt het systeem moeilijker. De consequentie van modificatie is dat de gebruikers op een andere manier getraind moeten worden en daarnaast zal er ook documentatie moeten plaatsvinden. Gebaseerd op de literatuur kan er door modificatie succes ontstaan en kan het een positief effect hebben op de Return On Investment (ROI). Daarnaast wordt modificatie toegepast om de weerstand van de werknemer te verkleinen. Als modificatie daarentegen wordt toegepast in een organisatie waar er een cultuur heerst tegen veranderingen, is de training voor de werknemers een uitdaging.

Als er een minor upgrade wordt gedaan heeft dit impact in de vorm van inspectie en het testen door de consument. Als er een major upgrade wordt toegepast, ontstaat er een nieuwe versie, waardoor de impact groter is. Als eerste kunnen de parameters worden aangepast. Hierdoor zal er een herconfiguratie moeten worden toegepast. Als tweede kan het datamodel veranderen. Als gevolg zal er een datamigratie plaatsvinden. Als derde zullen door de major upgrade extensies en modificaties opnieuw worden uitgevoerd. Als door de modificatie het datamodel is aangepast, maakt dit datamigratie risicovol en duur. Als er middelen van de leverancier worden gebruikt voor de migratie van software en data, dan moet de modificatie ongedaan worden gemaakt. De upgrade moet plaatsvinden waarna de modificatie opnieuw moet worden uitgevoerd. Door een nieuwe update kan het aantal veranderingen die door de klant zijn aangebracht verminderen en een nieuwe upgrade kan leiden tot een verminderd onderhoud in de toekomst.

De impact van een verandering in scope is hetzelfde als herconfiguratie, omdat de invulling van de parameters veranderen. Door een verandering van single-site naar multi-site zal de datastructuur worden veranderd, waardoor er datamigratie vereist is. Bovendien leidt een verandering in de scope ook tot een verandering in de organisatie.

Tot slot als er een verandering plaatsvindt in de deployment, dit kan bijvoorbeeld gebeuren door te veranderen van on premise (OP) naar ASP, kunnen de kosten afnemen. Als er wordt overgestapt op SaaS dan kunnen de kosten nog verder afnemen en kan er geen verdere modificaties plaatsvinden.

De impact van ERP op de technologie en de organisatie is weergegeven in tabel 3 op bladzijde 10 van de vijfde reader.

Vervolgens wordt erop ingegaan hoe van een ERP-systeem informatie kan worden afgeleid voor het management. Het ERP-systeem bevat veel informatie, waardoor het soms lastig is om data om te zetten in betekenisvolle informatie.

In een ERP-systeem wordt informatie nooit twee keer vermeld. Door gegevens onder te brengen bij een item wordt dubbele vermelding van gegevens vermeden. Als gevolg worden de gegevens benoemd als attribuut van een item. Dit kan weergegeven worden in een UML-diagram. Een voorbeeld is hiervan opgenomen in figuur op bladzijde 13 van de vijfde reader.

Als een overzicht van gegevens wordt teruggebracht naar een overzicht waarbij het dubbel benoemen van informatie vermeden wordt, noemt men dit normalization. Transactie systemen zijn er zoveel mogelijk op normalization ingesteld, want hierdoor kan de business logic objecttypes op een verfijnde manier ontwerpen.

Het kan ook voorkomen bij specialisatie. Door middel van een business logic kan een systeem aan de hand van bijvoorbeeld een Nederlandse postcode en huisnummer het adres weergeven. Zo'n business logic wordt ontworpen door middel van het object adres, waarbij adres in Nederland een specialisatie is van adres.

Door gedetailleerde specialization en normalization leidt dit tot veel objecttypes in een ERP-systeem, als gevolg is het soms lastig om alle informatie in een overzicht te weergeven.

Het tegenovergestelde van normalization is een Join van tabellen. De objecttypes in een UML class diagram kunnen worden gezien als de tabellen die worden weergegeven in een database. In het genoemde voorbeeld in de reader is de join van de tabellen order line en de tabel order opgesteld aan de hand van de foreign key kolom order ID in de tabel order line. De join houdt in dat voor elke orderregel (elke regel in de order line tabel) geselecteerde attributen van de order worden toegevoegd. Voorbeelden van deze attributen zijn de besteldatum, productnaam, hoeveelheid, klant en verkoper. Vervolgens kan met behulp van de foreign key product name een nieuwe join ontstaan voor item. Als gevolg heeft elke regel in de tabel waarden betreft de prijs per stuk en de kosten per stuk. Door deze twee join tabellen is het niet mogelijk om geheel tabel 1 (bladzijde 12) te kunnen vormen, omdat bepaalde gegevens nog ontbreken.

De gegevens die in tabel 1 zijn weergegeven zijn derived data. Een voorbeeld van een ontbrekend gegeven is de winst. De winst wordt bepaald aan de hand van de prijs en kosten per stuk en de hoeveelheid. De berekening kan worden gemaakt als onderdeel van de business logic of kan worden berekend als de figuur is afgeleid van het transactiesysteem. In ieder geval is het niet altijd zo dat direct vanuit de database de benodigde data kan worden afgeleid.

Nog een reden waarom de benodigde data niet direct afgeleid kan worden is omdat het transactiesysteem de gegevens van vandaag toont. Zo weergeeft het systeem alleen actuele data en geen historische data. Het is wel mogelijk om op te slaan welke life cycle een object heeft doorlopen. Maar als er bijvoorbeeld wordt gekeken naar figuur 1. Dit figuur komt uit 2010. Inmiddels kunnen bepaalde dingen niet meer verkocht worden of zijn de inkoop en verkoopprijzen gewijzigd.

Management maakt meestal een overzicht met gegevens over een bepaalde periode. Doordat actuele data wordt weergegeven is het transactiesysteem, wijzigen continue de gegevens. Als gevolg is het niet mogelijk om in het systeem historische data terug te halen.

Om hierop te anticiperen is het van belang om direct de gegevens uit het systeem te halen die nodig zijn voor het management. Als er weer wordt teruggegaan naar tabel 1. Stel in 2016 wil je een overzicht opstellen over de verkopen in 2010. Het is dan van belang om vanaf 2010 de juiste gegevens op te vragen. Als het namelijk in 2016 wordt gedaan is het erg lastig om dan nog de juiste kostprijs en de verkoopprijs te achterhalen. Bovendien zullen bepaalde verkopers niet langer meer in dienst zijn, bepaalde klanten zijn niet meer klant, enzovoorts.

Figuur 1 op bladzijde 12 van de vijfde reader wordt een data warehouse genoemd. Een data warehouse is toegespitst op een objecttype, zoals verkooporderregels en verzamelt door de tijd heen alle relevante gegevens. Een objecttype wordt verder uitgebreid door toepassing van meerdere join operations en relevante attributen toe te voegen. Voorbeelden hiervan in figuur 1 zijn de klant, verkoper, kostprijs en verkoopprijs. Maar ook wordt door de derived data het gegeven winst gegeven.

Verschillen tussen een datawarehouse en een transactiedatabase is hieronder in tabel 2 weergegeven.

Data warehouses	Transactional databases
Zijn genormaliseerd	Zijn niet genormaliseerd
Betreft meerdere objecttypes	Betreft een objecttypes
Kunnen veranderen	Wordt uitgebreid en data wordt opgeslagen, maar ondergaat geen veranderingen
Gaat over transacties	Gaat over het analyseren van data
Er wordt gewerkt aan een paar objecten	Meerdere objecten worden geanalyseerd
Derived data wordt elke keer berekend als gegevens nodig zijn	Derived data wordt berekend wanneer het object is opgeslagen en onveranderd blijft
Geeft actuele informatie	Geeft informatie over een bepaalde tijd

Tabel 2 – Verschillen tussen data warehouses en transactional databases

Data warehouses kunnen in verschillende vormen voorkomen. Dit kan bijvoorbeeld in een spreadsheet, wat vaak wordt gebruikt in klein- en in middenbedrijven. Een gespecialiseerde vorm van opslaan en analyseren van data is OLAP (Online Analytical Processing) en is het contrast van technologieën van een transactiesysteem wat vaak OLTP (Online Transaction Processing) wordt genoemd.

Als een data warehouse wordt gepresenteerd in OLAP, worden er dimensies aangemaakt voor het analyseren van data. In figuur zijn deze dimensies tijd, product, locatie van de klant en de verkopen van de organisatie. De dimensies bestaan uit een hiërarchie waarbij deze erg gedetailleerd of erg breed kan zijn. Een voorbeeld hiervan is bijvoorbeeld de tijd. De tijd kan worden weergegeven in dagen en maar ook in maanden.

Naast dat dimensies alleen voor een single data warehouse worden toegepast. Kunnen de dimensies ook bijvoorbeeld worden gebruikt voor het beantwoorden van de vraag wat de totale winst is van alle verkopers in een bepaald gebied van een bepaald jaar. Hierbij moeten de dimensies wel gedetailleerd zijn, want gegevens zoals de het jaar, producttypes per product etc. moeten worden verzameld.

Dimensies kunnen naast tabellen ook worden weergegeven in blokken. Een voorbeeld is hiervan opgenomen op bladzijde 17 van de vijfde reader.