

Software Engineering 1: Systeemanalyse en –Ontwerp (P1) Theorie

0. Inleiding

- Systeemanalyse = Informatie verzamelen, organiseren en interpreteren
- Analyst = Architect v software
- Informatiesysteemanalyse en –ontwerp = methode
 - Doel: het verbeteren van organisatorische systemen (meestal door toepassen van software waarmee WN's hun bedrijfstaken gemakkelijker en efficiënter kunnen uitvoeren)
 - Gestructureerde aanpak vereist
 - SDLC: Systems Development Life Cycle
 - Systems Planning and Selection (Vaststellen)
 - Systems Analysis (Analyseren)
 - Systems Design (Ontwerpen)
 - Systems Implementation and Operation (Implementeren)
- Kernbegrippen
 - Toepassingssoftware
 - Ter ondersteuning van specifieke organisatorische functies of processen (vb: voorraadbeheer, salarisadministratie, marktanalyse)
 - Doel: Omzetten van data (gegevens) naar informatie
 - Methodologieën vs technieken vs hulpprogramma's
 - Methodologieën: stap voor stap aanpak, welke weg volgen?
 - Technieken: Ondersteunende processen (interview, planning)
 - Tools: CASE-tools (Computer-Aided Software Engineering)
 - =Organisatorische aanpak: Gegevensgeoriënteerd vs procesgeoriënteerd
- Kenmerken ve informatiesysteem
 - Componenten (subsystemen)
 - Onderling gerelateerde componenten (afhankelijk van elkaar)
 - Begrenzing (blackbox)
 - Blackbox= Concept v doos waar je iets in kan steken en iets uit kan halen, maar niet weet hoe de doos intern werkt. Standaardbenadering user => software
 - Doel (bestaansgrond vh systeem)
 - Omgeving (buiten begrenzing, toch van invloed)
 - Interfaces (raakvlakken tussen systeem en omgeving)
 - Invoer (voedsel van buitenaf)
 - Uitvoer (terug naar omgeving)
 - Beperkingen (van binnenuit of van omgeving)
- Hoe informatiesystemen werken:

- Decompositie: Opsplitsen van systeem in kleinere subcomponenten
- Modulariteit: Als gevolg van decompositie
- Koppeling: De mate waarin subsystemen met elkaar samenhangen, moet zo zwak mogelijk zijn. Uitvallen van een systeem mag geen invloed hebben
- Cohesie: Aantrekkingskracht van processen van een subsysteem (moet sterk zijn)
- Procesgeoriënteerde aanpak
 - Richt zicht voornamelijk op wat het systeem moet doen
 - Manier waarop gegevens verplaatst en veranderd worden
 - Nadeel: Leidt dikwijls tot overtolligheid van gegevens (bepaalde gegevens nodig voor versch subsystemen)
- Gegevensgeoriënteerde aanpak
 - Gericht op ideale structuur van gegevens, onafhankelijk van
 - De plaats waar ze nodig zijn
 - De manier waarop ze gebruikt worden
 - Gegevensmodel(len)
 - Ontwikkeling ve database
- Moderne aanpak
 - Gegevens & Informatie = feiten, geschikt voor menselijke interpretatie
 - Gegevensstromen = gegevens die zich verplaatsen
 - Beschrijving van bronnen
 - Bestemming binnen de gegevensstroom
 - Proceslogica: Beschrijft de stappen die de gegevens veranderen
- Taken ve systeemanalyst
 - Bestuderen v problemen & ontdekken van oplossingen
 - Duidelijkheid scheppen in omschrijving aanpassingen
 - Communicatie naar programmeur verzorgen/vertalen
- Vaardigheden ve systeemanalyst
 - Goed analytisch denken om organisatie/functies te begrijpen
 - Inleven in het begrip systeem
 - Technische kennis om beperkingen juist in te schatten
 - Communicatieve vaardigheden om in team te functioneren
 - Kennis van allerhande programmeertalen, OS, hardwareplatformen
 - Managementvaardigheden

1. ERD

- ERD = Entity Relationship diagram (!= DB diagram)
- Basis building blocks
 - Entiteitstype (Soort object) Entiteitsvoorkomen(s)
 - Attribuuotype met attribuutwaarden
 - Subject area
 - Relatietype
- Technieken (Verskil ligt in symbolen): **Kraaienpoottechniek**, IDEF1X, Chen-notatie
- Entiteitstype (entity type)

- Groep van objecten die beschreven worden door dezelfde eigenschappen en die binnen het te ontwikkelen systeem als zelfstandig kunnen beschouwd worden
- Entiteitsvoorkomen (entity occurrence)
 - Uniek identificeerbaar object van entity type
 - Heeft levenscyclus (wordt gecreëerd, ondergaat statusveranderingen en wordt evt terug vernietigd)
- Grafische voorstelling
 - Naam = zelfstandig naamwoord in enkelvoud
 - In kader
- Relatietype (relationship type)
 - Associatie tss entiteitstypen
 - Krijgt naam die exact aanduidt wat functie is van die relatie
 - Koppeling (relationship occurrence of link)
 - Concreet verband tss precies één entiteitsvoorkomen van de deelnemende entiteitstypen
 - Grafische voorstelling
 - Naam = ww in TT enkv
 - Lijn die deelnemende entiteitstypen verbindt
 - Role names: wanneer er tss twee entiteitstypen meer dan één relatie kan gevonden worden
 - Graad van relatie = aantal deelnemende entiteitstypen. Meestal binair (2), soms ternair (3) of quaternair (4)
 - Recursieve relatie: entiteitstype met zichzelf
 - Cardinaliteit: 1, N. Hoeveel entiteitsvoorkomens van ent1 kunnen gekoppeld zijn aan 1 voorkomen van ent2
 - Modaliteit: verplicht of niet
 - M:N relatie vervangen door 1:N (intermediërende entiteit) M:1
- Attribuuttype
 - Kenmerk van entiteitstype
 - Attribuutwaarde
 - Beschrijft exact elk entiteitsvoorkomen van dat entiteitstype
 - Elk attribuuttype moet tijdens gegevensanalyse voldoende gedocumenteerd worden (datatype, lengte, verplicht of niet, domein (=verzameling v mogelijke waarden))
 - Enkelvoudige (atomaire) en samengestelde attributen (vb: adres)
 - Single-valued (één waarde voor één entity occurrence) of multi-valued (meerdere waarden voor elk entiteitsvoorkomen, min & max opgeven), vb: afdeling kan 1 tot 5 telnr's hebben
 - Sleutels
 - Alternate key (AK): minimale verzameling van attributen die ieder entiteitsvoorkomen op unieke wijze identificeren (bv: staat+nr+bus+postnr)
 - Primary Key (PK) = identifier: gekozen kandidaatsleutel om effectief een entiteitsvoorkomen uniek aan te duiden (vb: isbn-nr)
- Subject area
 - Interessegebied v onderneming
 - Verzameling v entiteitstypes rond één centraal entiteitstype

- Zelfstandig naamwoord in meervoud (meestal naam centraal entiteitstype).
Hoofdletters
- Alle entiteitstypen moeten in subject area zitten
- Een entiteitstype mag maar tot 1 subjectarea behoren (tenzij child subarea)

2. SDLC

- = System Development Life Cycle
 - Algemeen toegepaste methodologie
- “Klassieke” levenscyclus
 - Systeemplanning en –selectie
 - Systeemanalyse
 - Systeemontwerp
 - Systeemimplementatie en inbedrijfsstelling
- # en inhoud fasen staat niet vast en iedereen kan eigen systeem definiëren
- Fasen zijn niet perfect begrensd
- Spiraalmethode: De fasen komen steeds terug, met elke iteratie minder lang

2.1. Fase 1: Systeemplanning en –selectie

- Iemand stelt de behoefte aan nieuwe en/of uitgebreidere systemen vast
- Proces
 - Inzicht verkrijgen in “waarom een systeem moet worden gebouwd”
 - Waarde voor het bedrijf bepalen (kostenbesparend/winst)
 - Haalbaarheidsanalyse: technisch, economisch, organisatorisch
 - Management
 - Hoe moet projectteam te werk gaan om het systeem te bouwen
 - Projectwerkplan = beschrijving hiervan

2.1.1. Opzetten en selecteren van een projectplan

- Vaststellen/opstellen van (potentiële) ontwikkelprojecten door
 - Topmanager
 - Stuurgroep v managers
 - Gebruikersafdelingen (met hulp van analisten)
 - Ontwikkelteam of senior Information Systems manager

| Wie | Kosten | Duur | Complexiteit | Grootte | Doel |
|--------------------|---------|---------|--------------|---------|--------------------------------|
| Topmanager | Hoogst | Langst | Hoogst | Grootst | Strategisch (goedkeuring mgmt) |
| Stuurgroep | Hoog | Lang | Hoog | Groot | Cross-functional |
| Gebruikersafdeling | Laag | Kort | Laag | Klein | Afdelingsgericht |
| Ontwikkelteam | Onzeker | Onzeker | Onzeker | Onzeker | Integratie bestaande sys. |

- Identificatie van problemen
 - Vergelijk output met performantie-criteria
 - Teveel fouten
 - Werk (te traag, foutief, onvolledig, niet) uitgevoerd
 - Observeer het gedrag van medewerkers
 - Hoog absentisme
 - Lage job-voldoening

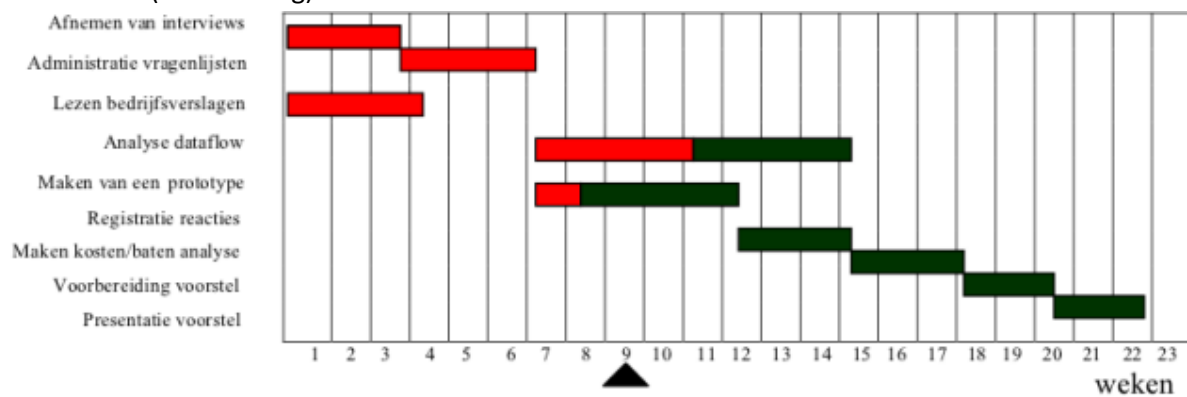
- Veel verloop
 - Luister naar externe feedback van (verkopers, klanten, leveranciers)
 - Klachten
 - Suggesties tot verbetering
 - Verlies van verkoop (markt)
 - Minder verkoop (aantal)
- Gebruikte criteria voor classificeren en rangschikken projecten
 - **Waardeketenanalyse:** Mate waarin activiteiten waarde en kosten vermeerderen bij het ontwikkelen van P/D. Proj met grootste algemene voordelen krijgen prioriteit
 - **Strategische richting:** Mate waarin project gezien w als hulp vr organisatie bij bereiken (strategische, lange termijn) doelen
 - **Potentiële voordelen:** Mate waarin proj gezien w als vergroting winst, verbetering service en duur vd voordelen
 - **Beschikbaarheid middelen:** # en type middelen die vr proj nodig zijn en hun beschikbaarheid
 - **Projgrootte/duur:** # personen en tijd nodig om project af te ronden
 - **Tech moeilijkheid & Risico's**
- Selecteren v proj en nemen v beslissingen
 - Mogelijke uitkomsten: accepteren, verwerpen, vertragen, focus verleggen, extern kopen, heranalyseren

2.1.2. Projecthaalbaarheid vaststellen

- FIG (Feasibility Impact Grid) = effect van uitvoerbaarheid, impact op (proces-, bedrijfs-)doelstellingen
- Mogelijke procesdoelstellingen
 - Versnellen proces
 - Eliminatie v onnodige of overbodige stappen
 - Combinatie v processen
 - Verminderen input-errors
 - Verminderen meervoudige (oplag, output)
 - Verbeteren v integratie tss systemen & subsystemen
- Mogelijke bedrijfsdoelstellingen
 - Vergroten winst
 - Ondersteuning bedrijfsstrategie
 - Verbeteren samenwerking met verkopers & partners
 - Verbeteren interne (samen)werking
 - Verbeteren interne besluitvorming
 - Verbeteren klantenservice
 - Verhogen werkvoldoening bij medewerkers
- Haalbaarheidsanalyse
 - Technisch: kunnen we het
 - Economisch: Moeten we het
 - Organisatorisch/Operationeel: Als we het bouwen, zal het gebruikt w?

2.1.3. Systeemontwikkelprojecten initiëren en plannen

- Activiteiten die plaatsvinden tijdens projectplanning
 - Draagwijdte, alternatieven en haalbaarheid beschrijven
 - Project verdelen in hanteerbare taken
 - Schatting v middelen maken en middelenplan opstellen
 - Eerste schema ontwikkelen
 - Communicatieplan ontwikkelen
 - Projectnormen en –procedures bepalen
 - Risico's vaststellen en inschatten
 - Voorlopig budget opstellen
 - Werkoverzicht maken
 - Basisprojectplan opstellen
- Grafische voorstellingen
 - GANTT chart (= eenvoudig)



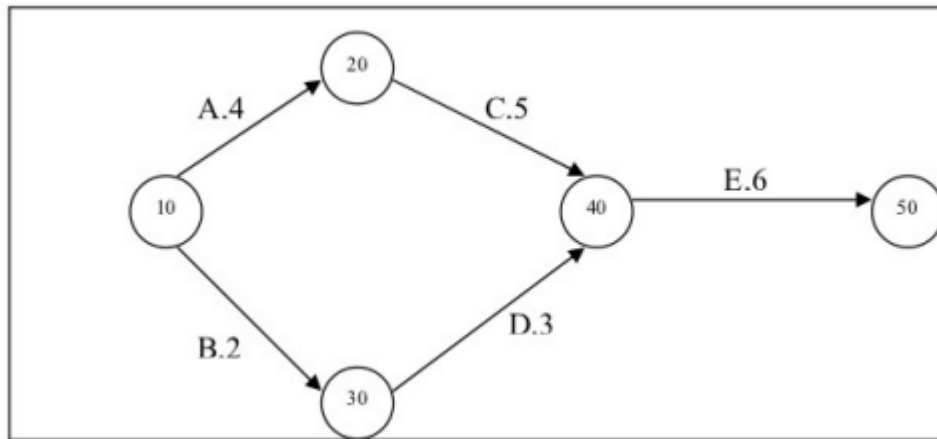
- Laat duur taken zien
- Laat tijdsoverlapping zien
- Laat (soms) beschikbare speelruimte zien binnen vroegst mogelijke startdatum en uiterste einddatum

- Netwerkdigramma



- Laat volgordeafhankelijkheid v taken zien
- Laat gn tijdsoverlapping zien, wel welke taken parallel kunnen uitgevoerd w
- Toont speelruimte door geg binnen activiteitsvierkantjes

- PERT diagram : om tijdskritisch pad op te sporen



2.1.4. Basisprojectplan opstellen en beoordelen

- Basisprojectplan opstellen & beoordelen
 - 4 belangrijke delen
 - Inleiding
 - Systembeschrijving
 - Haalbaarheidsonderzoek
 - Managementkwesities
 - Moet goedgekeurd w alvorens naar fase 2 te kunnen

2.2. Fase 2: Systemanalyse

- Antwoorden zoeken op vragen:
 - Wie gaat het systeem gebruiken?
 - Wat moet het systeem doen?
 - Waar en wanneer zal het systeem worden gebruikt?
- => Leidt tot analysestrategie
- Verzamelen van commerciële eisen: interviews, vragenlijsten
 - Ontwikkeling van eisenpakket: use-case, procesmodel, gegevensmodel
 - = Systemvoorstel of gevalstudie

2.2.1. Systemvereisten bepalen

2.2.1.1. Interactieve methoden

2.2.1.1.1. Interviews

- Voorbereiding
- Typen vragen
 - Open vraagstelling
 - Gesloten vraagstelling
 - Follow-up vragen

| Open vraag | | Gesloten vraag |
|------------|--------------------------------|----------------|
| Laag | Betrouwbaarheid vd data | Hoog |
| Laag | Efficiënt tijdsgebruik | Hoog |
| Laag | Accuraatheid vd data | Hoog |

| | | |
|----------|-------------------------------------|-------------|
| Veel | Diepgang | Weinig |
| Veel | Interviewvaardigheid vereist | Weinig |
| Moeilijk | Gemak van analyseren | Gemakkelijk |

- Vraagvolgordes
 - Piramide-structuur: Begin met concrete vraag en eindig met open vraag
 - Trechter-structuur: Begin met algemene vraag en eindig met gesloten vraag
 - Diamant-structuur: Concreet => Open => Gesloten

2.2.1.1.2. Vragenlijst

- Doel: inzicht krijgen in attitudes, de realiteit, het gedrag en de karakteristieken van verschillende belangrijke personen in de organisatie.
- Gesloten vragen kan je automatiseren
- Gebruiken als
 - Betrokken personen zijn geografisch verspreid
 - Aantal betrokken personen is groot
 - In vooronderzoek en wil algemene opinie kennen voor project in bepaalde richting te sturen
 - Wil zeker zijn dat alle problemen met huidig probleem ontdekt zijn

| Interviews | Eig | Vragenlijsten |
|-----------------------------------|---|---------------------------------------|
| Hoog | Informatierijkdom | Gemiddeld – Laag |
| Hoog – Gemiddeld | Benodigde tijd | Gemiddeld – Laag |
| Kunnen hoog zijn | Kosten | Gemiddeld |
| Goed: interviewer & geïnterviewde | Kans op vervolg & doorvragen | Beperkt: pas na verzamelen antwoorden |
| Duidelijke betrokkenheid | Vertrouwelijkheid | Geen betrokkenheid |
| Beperkte aantallen | Potentiële doelgroep | Grote aantallen |

2.2.1.1.3. Observatie

- Mensen kunnen soms moeilijk precies formuleren wat ze doen en hoe
- => Mensen observeren

2.2.1.1.4. Documentanalyse

- In documenten info te vinden over
 - Problemen met bestaande systemen
 - Opportuniteiten om aan nieuwe behoeftes te voldoen
 - Organisatorische richting die systeemvereisten kan beïnvloeden
 - Titels en namen v belangrijke personen
 - Waarden van organisatie of van individuen die kunnen helpen prioriteiten te stellen
 - Afwijkingen van normale gang van zaken
 - Reden waarom huidige systemen zijn zoals ze zijn
 - Bedrijfsregels die moeten opgelegd w door het informatiesysteem

2.2.1.2. Moderne methoden

2.2.1.2.1. JAD-sessie (Joint Application Design)

- Afnemen van interviews kost vrij veel tijd en informatie is vrij onbetrouwbaar.
- In een JAD-sessie worden gebruikers, managers en systeemontwikkelaars samengebracht voor reeks intensieve, gestructureerde bijeenkomsten die geleid worden door JAD-sessieleider
- Gebruiken als
 - Gebruikers iets nieuws willen, geen standaardoplossing
 - Bedrijfscultuur gezamenlijk oplossen van problemen steunt
 - Analisten voorspellen dat persoonlijke interviews minder ideeën zouden voortbrengen
 - Het bedrijf zijn personeel gedurende enkele dagen kan vrijmaken voor JAD-sessie
- Alle deelnemers moeten ervoor open staan en moeten zich betrokken voelen
 - Minstens één analist, vooral passieve rol.
 - Vertegenwoordiging van gebruikers (alle niveaus)
 - Leider is communicatiemanager.
 - 1 of 2 waarnemers met voldoende technische bagage
 - Verslaggever
- Waar? Buiten het bedrijf, aangename omgeving, zodat deelnemers door niets kunnen worden afgeleid.

| + | - |
|---|---|
| <ul style="list-style-type: none"> • Tijdsbesparend • Gebruikers voelen zich nauw betrokken • Meer kans dat ideeën gebruikers ook in uiteindelijk systeem zitten. • Kan nieuwe ideeën naar voren brengen (lijkt op brainstorming) | <ul style="list-style-type: none"> • Deelnemers moeten gedurende langere tijd vrij zijn • Slechte voorbereiding of onvolledig follow-up rapport & einddocumenten doet sessie in waarde afnemen • Kan nieuw zijn voor bedrijf |

2.2.1.2.2. Prototyping

- Bouwen van kleinere maar werkende versie van het gewenste informatiesysteem. Kan met behulp van CASE-tools
- Geschikt om informatiebehoefte bij de gebruiker te weten te komen
- Kan helpen
 - Als gebruikseisen niet voldoende zijn
 - Meerdere gebruikers en andere belanghebbenden bij het systeem betrokken zijn
 - Complexe ontwerpen uitwerking vragen om goede evaluatie te bekomen
 - Mogelijke communicatieproblemen tss analist en gebruiker
 - Nood aan snel werkend systeem

| + | - |
|---|--|
| <ul style="list-style-type: none"> • Eindgebruikers nauw betrokken • Direct inspelen op opmerkingen gebruikers • Ontwikkeling v systeem dat niet werkt kan vroeg stoppen | <ul style="list-style-type: none"> • Degelijke documentatie schrijven w vaak vergeten waardoor moeilijker werkend geheel kan ontwikkeld w • Kan ongebalanceerd zijn: teveel wensen |

| | |
|--|---|
| <ul style="list-style-type: none"> • Resultaat is systeem dat zeer nauw aan gebruikerseisen en –verwachtingen voldoet | <p>gebruiker X, mnder anderen</p> <ul style="list-style-type: none"> • Prototypen w vaak bekeken als stand-alone systemen, waardoor integratie met bestaande systemen verloren gaat • Sommige SDLC-controles w overgeslagen waardoor systeemvereisten kunnen vergeten w • Soms zien analisten & gebruikers dit als eindsysteem |
|--|---|

2.2.1.2.3. RAD (Rapid Application Design)

- Objectgeoriënteerde aanpak van systeemontwikkeling
- Omvat zowel ontwikkelmethodologie als software tools
- Leunt dicht aan bij prototyping
- 3 fasen
 - Bepalen systeemvereisten
 - RAD Design workshop: Gebruikers reageren op effectief werkend prototype & analisten verfijnen modules
 - Implementatiefase
- Wanneer gebruiken
 - In projectteam is er ervaring met deze methodologie
 - Specifieke bedrijfsredenen om deel van applicatieontwikkeling te versnellen (bv. Eerste zijn of gebruikers nauw betrokken bij bedrijfsdoelstellingen)
- Let op voor te veel versnelling

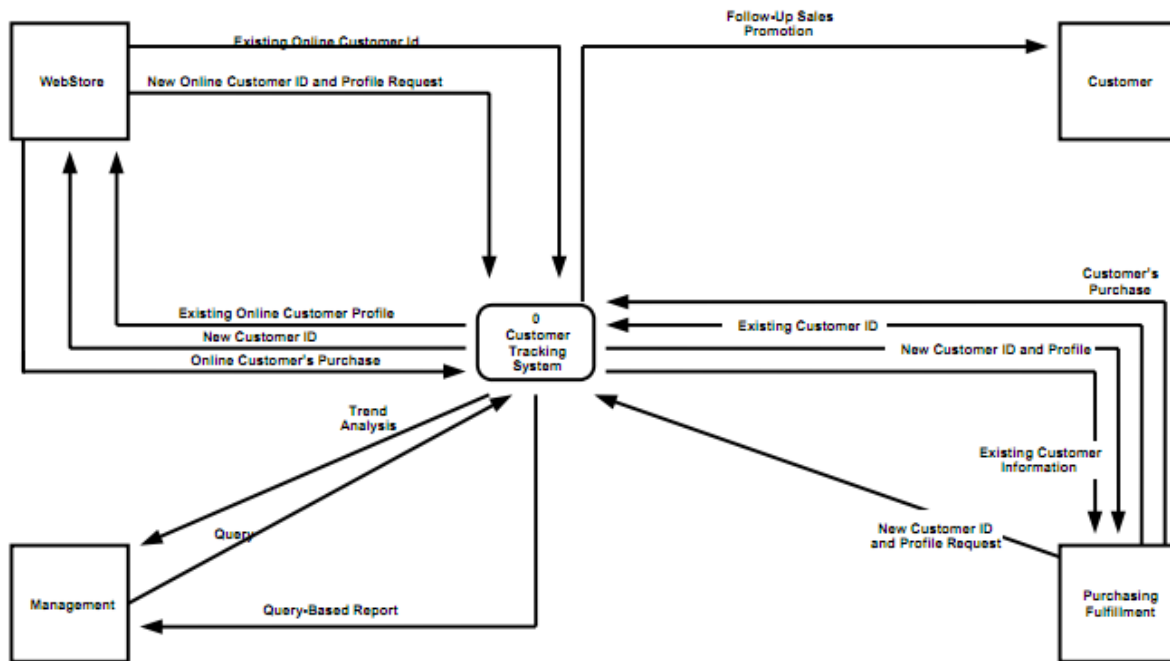
2.2.2. Systeemvereisten structureren

- Logica modelleren
- Gegevens modelleren (ERD)

2.2.2.1. Procesmodellering

- Grafisch voorstellen van processen/handelingen die gegevens verzamelen, bewerken, opslaan
- Veelgebruikte vorm is gegevensstroomschema (DFD: Data Flow Diagram). Laat verplaatsing van gegevens tss externe entiteiten, processen en gegevensoplagplaatsen

PWM : klantopvolgsysteem
Context Diagram



2.2.2.2. Logica modelleren

- Processpecificaties w opgebouwd voor elementaire processen op DFD. Hierop vind je beslissingslogica en formules die inputdata vh proces zullen omzetten in output data
- Doelstellingen
 - Dubbelzinnigheid vh proces verminderen. Analist begrijpt hierdoor beter hoe proces moet werken
 - Verkrijgen van precieze beschrijving v wat het proces moet doen. Dient later als uitgangspunt vr programmeur
 - Valideren v systeemontwerp. Krijgen extra controle of proces wel alle input krijgt die nodig is om output te leveren

Proces specificatie

Nummer proces : 1.3

Naam proces: *bepaal voorraad van het artikel*

Beschrijving: *bepaal of een artikel voldoende momenteel in voorraad is. Indien niet, plaats je het artikel in backorder. Bepaal de actuele voorraad.*



Input Data Flow:

Geldig artikelnr vanuit proces 1.2

Actuele voorraad vanuit het artikel bestand.

Output Data Flow:

Artikelnr en verkochte hoeveelheid naar proces 1.4 en proces 1.5

Het eventuele backorder naar Voorraad Controle.

Type van het proces:

Online

Batch

Manueel

naam subroutine/functie:

Proces logica:

If voorraad is greater than aantal besteld

Then move aantal besteld naar beschikbare hoeveelheid

move order artnr naar beschikbaar artikel

Else

Subtract voorraad van aantal besteld, resulterend in aantal backorder

Move aantal backorder to backorder artikel record

Move artnr naar backorder artikel record

Do write backordered record

Move voorraad naar beschikbare hoeveelheid

Move order artnr naar beschikbaar artikel

Endif

Verwijst naar : Naam :

Structured English

Beslissingstabel

Beslissingsboom

Onopgeloste issues : *Moet de hoeveelheid van dit artikel dat op de bestelling staat volledig in rekening gebracht worden?*

2.2.3. Kiezen van beste ontwerpstrategie

- Hardware & infrastructuur
- Alternatieve strategieën
 - Functionaliteit vs kosten
 - Systemeisen: verplicht / essentieel / gewenst
- Software-evaluatie
 - Outsourcing
 - Zelf ontwikkelen
 - Off-the-shelf: bestaande software aankopen
 - ERP (Enterprise Resource Planning software): volledige oplossing, alle data en processen in 1 systeem
 - Software op maat
- Evaluatie bestaande hardware & systeemsoftware
- Aanduiden meest kansrijke alternatief

| + | - |
|---|--|
| In House | |
| <ul style="list-style-type: none"> • Concurrentieel voordeel • Rechtstreeks betrokken bij bedrijf • Onderhoud door eigen personeel | <ul style="list-style-type: none"> • Beginkosten kunnen hoog oplopen • Evt inhuren of werken met ontwikkelteam • Zelf verantwoordelijk vr onderhoud |
| Off-the-Shelf | |
| <ul style="list-style-type: none"> • Constante verfijning & verbetering • Sterk betrouwbaar • Verhoogde functionaliteit • Meestal lage initiële kost • Reeds elders in gebruik • Training en help meegeleverd | <ul style="list-style-type: none"> • Niet geënt op bedrijf • Beperkt aanpasbaar • Toekomst leverancier onzeker • Minder betrokkenheid eigen personeel |
| ERP | |
| <ul style="list-style-type: none"> • Volledig geïntegreerd systeem • Basissysteem uitbreidbaar met modules | <ul style="list-style-type: none"> • Complex • Duur • Evt aanpassing van bedrijf aan systeem nodig |
| Software op maat | |
| <ul style="list-style-type: none"> • Bedrijven kunnen corebiz centraal houden, zonder te moeten focussen op IT • Geen verkwisting v personeelsuren aan niet-essentiële IT opdrachten | <ul style="list-style-type: none"> • Verlies v controle over data, systemen, IT medewerkers, planning • Doorlichten software house nodig • Gn strategisch voordeel meer |

| Producent | Wnr kiezen? | Interne personeelseisen |
|-------------------------|--|---|
| In house | Voldoende personeel & middelen & sys moet van nul af aan gebouwd w | Zeker nodig, kan variëren |
| Off-the-Shelf | Als ondersteunende taak generiek is | Voor het vaststellen van eisen en het evalueren v softwarepakketten |
| ERP | Volledige systemen die over interne grezen heen gaan | Mogelijk, maar voornamelijk consultants |
| Software op maat | Interne ontwikkeling onmogelijk maar vereisten zijn bedrijfseigen | Mogelijk, afhankelijk v toepassing |

2.2.4. Aanpassen BPP

- Volgens nieuwe plannen

2.3. Fase 3: Systeemontwerp

- Bepaalt de werking vh systeem n termen van hardware, software, netwerk, UI, databases, files, ...
- Architectuurontwerp (hardware, software, netwerkinfra)
- Interface-ontwerp (UI, GUI)
- Gegevensopslagontwerp (DB, files)
- Programma-ontwerp
- => systeemspecs => haalbaarheidsanalyse (fase 1) => management
- = logische structuren omzetten in fysische specificaties

2.3.1. Output-ontwerp

- Genereer output die zinvol is

- Hou zoveel mogelijk rekening met gebruiker
- Geen overload aan informatie
- Output moet op tijd geleverd w
- Kies juiste vormvoor output

2.3.2. Input-ontwerp

- Barcode-reader
- OCR

2.3.3. Database-ontwerp

- Zie hoofdstuk ERD

2.3.4. Interface-ontwerp

- Prototyping
- Papieren formulieren
- Navigatie
- Groepeer gegevensvelden in logische categorieën
- Geen input => Ontoegankelijk maken vr gebruiker
- Flexibiliteit
- Consistentie

2.4. Fase 4: Systeemimplementatie en onderhoud

- Daadwerkelijke bouw/aanschaf van het systeem
 - Meestal langste en duurste onderdeel

2.4.1. Implementatie db

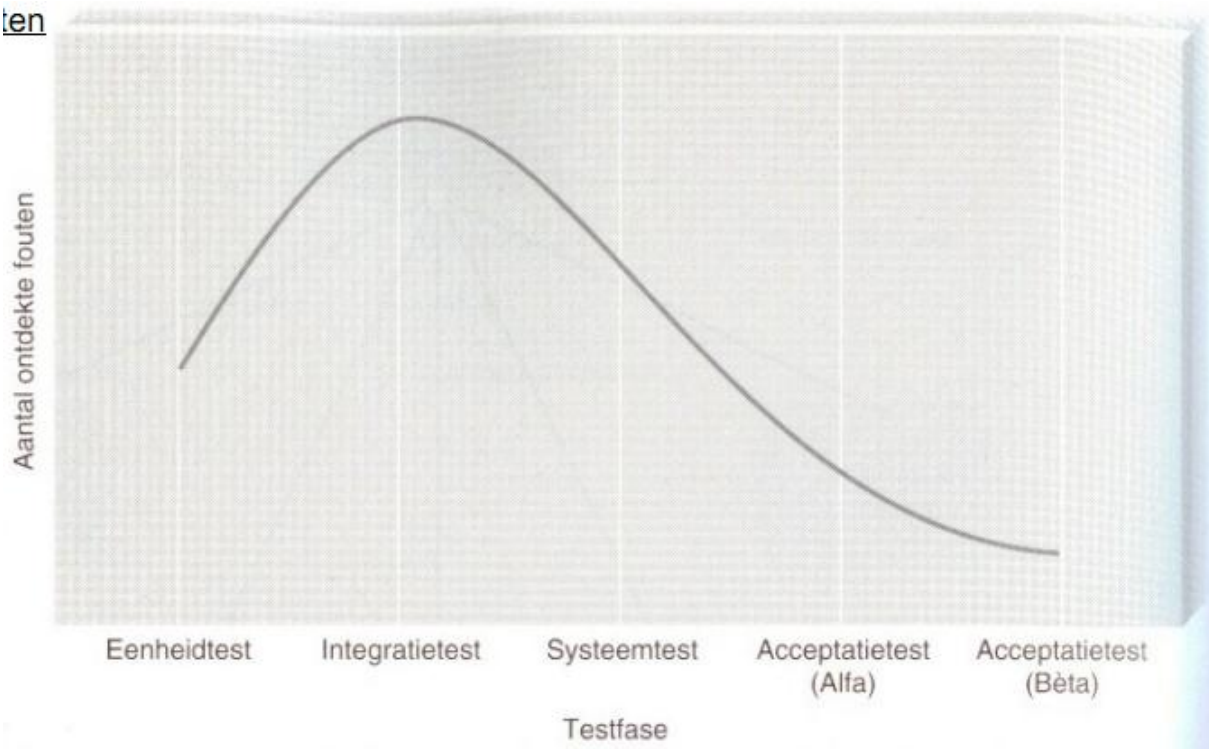
- Ook data overzetten

2.4.2. Coderen, testen, installeren

2.4.2.1. Testen

- 4 testfasen
 - Eenheidstests: testen van programmamodule met specifieke funcite
 - Blackbox-tests: programma-specificatie testen
 - Whitebox-tests: programma-broncode testen
 - Integratietests: Testen op samenwerken v modules, UI, use case, dataflow, systeeminterface
 - Systeemtests: testen op samenwerken v programma's in systemen = samenwerken v alle modules & programma's (test ook documentatie)
 - Acceptatietests: uitgevoerd door gebruikers (is het systeem klaar, voldoet het aan behoeften, is het acceptabel voor gebruikers), 2 fasen(alfatests met nepgegevens, betatests met echte gegevens)

ien



2.4.2.2. Installeren

- Oud systeem vervangen door nieuw systeem (incl conversie gegevens)
- 3 mogelijkheden
 - Directe installatie: oud systeem ineens vervangen door nieuw (=goedkoopst). Nadeel: risico + (mogelijk) grote invloed op organisatie
 - Parallele installatie: oud & nieuw bestaan naast elkaar (=veilig). Nadeel: dubbel werk & duur
 - Gefaseerde installatie: stapsgewijze overgang oud => nieuw. (=spreiding v risico in tijd). Nadeel: twee systemen naast elkaar + extra programmeerwerk (koppeling)

2.4.3. Risicobeleid

- Doel: risico's zo klein mogelijk houden
- 3 alternatieven
 - Preventieve maatregelen nemen
 - Verzekering nemen of financiële reserve aanleggen
 - Risico aanvaarden (zonder maatregelen te nemen)
- Meestal afweging kosten ⇔ beveiligingspercentage

2.4.4. Documenteren

- Systeemdokumentatie: vertrouwd maken met technische kant (vb: voor onderhoud)
- Gebruikersdocumentatie: beschrijving v werking/gebruik v systeem
- Een goede documentatie betekent besparing op het vlak v training & ondersteuning

2.4.5. Training & ondersteuning

- Verschillende types trainingsmethoden
 - Door aanwezige expert
 - Computerondersteunende functies

- Geven v vormingscursussen
- Met behulp v software-componenten
- Tutorials
- Interactieve trainingsmanuals
- Via externe bronnen zoals leveranciers
- Gebruikersondersteuning
 - Online / Helpdesks / Informatiecentra
 - Herstel/Backup mogelijkheden
 - Nieuwsbrieven
 - Gebruikersgroepen

2.4.6. Systeemonderhoud

- Langste fase voor een systeem: nieuwe eisen, presentatieverbetering
- Typen v onderhoud
 - Corrigerend onderhoud = herstellen & verbeteren v fouten en gebreken. Gn nieuwe functionaliteit. Meestal dringend
 - Adaptief onderhoud = sys aanpassen aan gewijzigde omstandigheden. Wel nieuwe functionaliteit. Meestal minder dringend.
 - Perfectionerend onderhoud: Prestaties/kwaliteit v systeem verhogen
- Meten
 - Mean Time Between Failure (moet toenemen in tijd)
 - Mean Time To Repair