

H02: eliciteren

Processen ontdekken

- opzetten van het project: samenstellen team
- Informatie verzamelen: bekomen van goed begrip vh proces
- Begeleiden van de modelleertaak: procesmodel
- Kwaliteitsborging: modellen beantwoorden aan kwaliteitscriteria

Uitdagingen

- Gefragmenteerde proceskennis
- Domein-experten denken op 'instance'-niveau
 - Proces-instance: unieke passage doorheen een proces → voor 1 bepaalde klant/leverancier/...
- Kennis over procesmodellering is zeldzaam

Voorbeeld van een proces: lassen van een reeks identieke stukken op een lasrobot

Expertise van procesanalisten

- Problemen begrijpen
 - Kennis vh probleemdomein
 - Kennis vd organisatie helpt om probleem te structureren
- Problemen oplossen
 - Identificatie procestriggers
 - Hypothese-beheer: formuleren en testen van hypotheses
 - Uitzetten van doelstellingen: wat is de volgende stap
 - Top-down strategie gebaseerd op de doelstellingen
- Modelleerskills
 - Structuur, duidelijke layout, systematische labels
 - Expliciete start- en eindpunten
 - Gepaste granulariteit & decompositie

Elicitatietechnieken

- Creatieve technieken
 - Brainstorm
 - Invalshoek veranderen
 - Gebruik maken van een analogie
- Uitvraagtechnieken
 - Interviewen
 - Enquête houden
- Observatietechnieken
 - Veldobservatie
 - Werkstage

- Documentatie-georiënteerde technieken
 - Systeemarcheologie
 - Lezen vanuit specifiek oogpunt
 - Hergebruik van requirements
- Ondersteunende technieken
 - Mind mapping
 - Workshops
 - CRC-kaarten
 - Audio & video opnamen
 - Use cases
 - Prototypen

Creatieve technieken

Brainstorm

- Nadelen
 - Ervaren moderator nodig
 - Sommige deelnemers veel dominanter dan anderen
- Alternatieven: brainwriting via 6-3-5 methode of GPS

Invalshoek veranderen: 6 denkhoeden

- Blauw: overzicht, controle, afstandelijkheid, voorzitter, dirigent
- Zwart: kritiek, analyse, negatief, zwartkijken, oordelen
- Rood: emoties, gevoelens, ingevingen, intuïtie, spontaan
- Wit: feiten, cijfers, informatie, objectiviteit, neutraal
- Groen: creativiteit, alternatieven, provocatie, energie, verandering
- Geel: positief, constructief, voordelen, kansen, optimisme

Gebruik maken van een analogie

- Analogieën stimuleren creativiteit
- Bv.: ziekenhuislogistiek rond behandelen van patiënten vergelijken met werkprocessen op Zaventem

Uitvraagtechnieken

Interviewen

- Meest gebruikte techniek
 - Huidige uitdagingen, ideale toekomstbeeld, ...
 - Verstaan & ontdekken van details
- Doel van interview kan verschillen
- Succesvol interviewen
 - Voorbereiding & ervaring
 - Verstandhouding opbouwen
 - Juiste persoon interviewen
 - Bereidheid interviewee

- Sterkten
 - Focus op dialoog
 - Uiten van gevoel & opinie in privé sfeer
 - Opbouwen ve relatie
 - Discussie zonder de nadelen ve meeting
- Nadeel
 - Veel tijd in beslag door feedbacksessies: interview → documentatie/modellen → verificatie → validatie → interview → ...
- Challenges
 - Resultaat kan verkeerd worden geïnterpreteerd
 - Soms wordt info achtergehouden
- Waarom interviewen?
 - Achterliggende reden voor een feature/upgrade/...?
 - Is er een bug die gefixed moet worden?
 - Wie wordt er geïmpacteerd?
- Wie interviewen?
 - Sponsor: leadership perspectief → inzicht in visie & uiteindelijke doel
 - Manager & verantwoordelijken: impact op hun omgeving
 - Eindgebruikers: direct inzicht in problemen/opportunities, huidige frustraties
- Hoe interviewen
 1. Voorbereiding
 - Doel, rollen, vragen, fysiek/online
 - Interview plan: datum, plaats, project, interviewee, rol/reden vd interviewee, hoofdvragen
 - Vragen: 3 tot 5 hoofdvragen, open vragen en gesloten vragen (enkel voor verduidelijking) → 3-5 open in combi met gesloten
 2. Uitvoeren interview
 - Band opbouwen, verwachtingen stellen
 - Goeie vragen stellen
 - Zoek wat nodig is, niet gewenst: stated vs actual requirements
 - Actief luisteren
 - Be attentive
 - Ask open-ended questions
 - Ask probing questions
 - Request clarification
 - Paraphrase
 - Be attuned to and reflect feelings
 - Summarize
 - Notuleren
 - Afsluiten vh interview
 3. Interview - follow-up
 - Formele bedanking: bedanken voor tijd
 - Nota's review: bedoelingen juist gecapteerd?
 - Acties opstellen

Enquête houden

- Fysiek of digitaal
- Open of gesloten, of combinatie
- Voordelen
 - Veel info in korte tijd
 - Snelle verwerking
 - Geen remmingen indien anoniem
- Nadelen
 - Deelnemers kunnen vragen verkeerd interpreteren
 - Je mist de non-verbale communicatie

Observatietechnieken

Veldobservatie

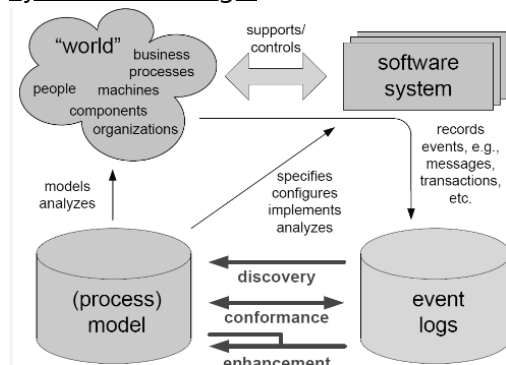
- Gebruiker werkt in bijzijn van analist
- Varianten: geen interactie of gebruiker legt uit wat hij doet

Werkstage

- Analist voert zelf het werk uit
- Tijdrend, levert diepgaande kennis op

Documentatie-georiënteerde technieken

Systeemarcheologie



Lezen vanuit specifiek oogpunt

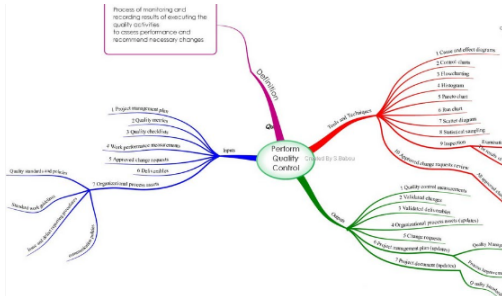
- Documenten verwijzen naar rollen, activiteiten business-objecten
- Formele documentatie
 - Organigram van de organisatie
 - Tewerkstellingsplannen
 - Kwaliteitsrapporten(bv. bijcertificaties), auditrapporten
 - Interne policy: missie-, visie-, strategiedocumenten
 - Interne en externe reglementen
 - Woordenlijsten en handboeken
 - Workinstructies
 - Invulformulieren

Hergebruik van requirements

- Eerder uitgevoerde analyses
- Tijd & kosten voor analyse sterk reduceren

Ondersteunende technieken

Mind mapping



Workshops

- Key-stakeholders samenbrengen
- Discussiëren om gedeeld begrip te creëren
- Software-ondersteund
 - Afzonderlijke rol naast moderator
 - Tijdens workshop modellen tekenen
- Model dient als referentie voor verdere discussies
- Alternatief (voor SW): brown-paper workshops (met post-its en tijdlijn)

CRC-kaarten

- Class Responsibility Collaboration
- Relevante business-objecten op kaarten schrijven
- Workshopdeelnemers voegen daar eigenschappen aan toe
- Kaarten gebruikt om processen & requirements in kaart te brengen

Student	
Student number	Seminar
Name	
Address	
Phone number	
Enroll in a seminar	
Drop a seminar	
Request transcripts	

Audio & video opnamen

- Gebruikt bij veldobservatie, interviews & workshops
- Nadeel: deelnemers gaan zich misschien anders gedragen

Use cases

- Elementaire bedrijfsprocessen: 1 persoon, 1 tijdsspanne, 1 plaats
- Deel van een groter geheel
- Helpen om elicitatieproces te structureren

Prototypen

= werkende software van kritieke delen ve toekomstig systeem

Techniekkeuze

- Combinatie van technieken nodig
- Houd bij keuze rekening met
 - Menselijke aspecten
 - Communicatieve & persoonlijke vaardigheden van stakeholders
 - Ervaring met bepaalde technieken
 - Mate waarin stakeholders zich bewust zijn van requirements
 - Organisatorische aspecten
 - Beschikbaarheid belanghebbenden: veldobservatie goed voor weinig tijd
 - Beschikbare budget & doorlooptijd
 - Workshop is tijdbesparen, letten op geografische spreiding
 - Creatieve technieken minder gangbaar bij fixed-price-date
 - Vakinhoudelijke aspecten
 - Vereiste detailniveau beïnvloedt keuze
 - Ervaring analist met bepaalde technieken

Verskil in keuze van technieken

- Startup met 100 werknemers vereist andere aanpak dan overheidsbedrijf

Invloed van de bedrijfscultuur

Open cultuur

- Alle medewerkers aangemoedigd om ideeën te delen
- Gebruik workshops om ideeën te delen

Strikt-hiërarchische organisaties

- Zorg dat elke stakeholder gelijk aan bod komt
- Anonieme enquêtes als aanvullende techniek

Addendum: interview tips

- Gestructureerd vs ongestructureerd
- Combinatie open en gesloten vragen
- Veronderstelling: analist en stakeholder verstaan elkaars terminologie
- LSD: Luisteren, Samenvatten, Doorvragen

Luistervaardigheden

- Empathie: spreker op een ondersteunende & behulpzame manier uithoren
- Analyse: concrete info zoeken en feiten & gevoelens uit elkaar houden
- Synthese: stuur gesprek pro-actief in richting van vooraf vastgestelde doelstelling

Enkele tips

- Planning
- Goede voorbereiding: info zoeken over persoon/bedrijf, vragen voorbereiden
- Tijdens gesprek
 - Verwelkoming, neem plaats
 - Begin met neutrale vragen
 - Veel-prater: gesloten/half open vragen
 - Niet-prater: open vragen
 - Bedanking aan einde gesprek
- Non-verbale communicatie: let op houding

H03: documenteren

Situering



Functionele requirements beschrijven

- Use cases: analyse techniek & specificatie
- User stories: communicatiemiddel voor requirements & moet aangevuld worden met extra technieken

User stories

Vorm

- Als een {wie}
- Wil ik {wat}
- Zodat {waarom}

Wat

- Korte beschrijving van wat een gebruiker wil
- Beschrijft in gewone spreektaal van de gebruiker wat de gebruiker (moet) doen
- Gebruikt binnen agile software development
- Weinig details, moet passen op een post-it

Waarom

- Makkelijk te verstaan: geen voorkennis vereist of technisch jargon
- Makkelijk te delen: doel is communicatie tussen IT & business
- Lage inspanning: snel opstellen, kan just-in-time
- Moeten niet perfect zijn: discussie zal voor duiding zorgen
- Makkelijk te plannen: stelt een werkitem voor

Eigenschappen

- Niet gebonden aan IEEE-guidelines
- Zijn geen use cases: scope, volledigheid, levensduur, doel

Richtlijnen → INVEST

- Onafhankelijk → Independent
- Onderhandelbaar → Negotiable
- Waardevol voor gebruiker & opdrachtgever → Valuable
- Schatbaar → Estimatable
- Klein → Small
- Testbaar → Testable

Negotiable - onderhandelbaar

- Beschrijven requirements zonder te specificeren
- Start de conversatie
 - Product owners: bepalen richting (wat)
 - Developers: hoe er te maken (hoe)
- User stories kunnen wijzigen doorheen de tijd
 - Meer kennis
 - Stories splitten of worden onnodig
- Zo laat mogelijk details toevoegen

Valuable - waardevol

- Moeten business waarde bevatten
 - Meer opbrengsten, lagere kosten, klanten aantrekken, personeel efficiënter maken, ...
- Gesloten stories: eindigt met een user's goal
- Slice the cake: aandacht geven aan alle lagen van de applicatie
- Database tabellen aanmaken, business logica schrijven, UI... zijn onderdelen van een user story, geen user story op zich
- Voorbeeld
 - Als een klant wil ik producten kunnen filteren op prijs zodat ik snel de producten kan vinden binnen mijn prijsklasse

Stories, epics & tasks



- Meerdere niveau's van detail: business ↔ technisch

Epics

- Groot stuk functionaliteit die business wil hebben & waarde oplevert
- Opgeleverd ahdv kleinere user stories, meerdere iteraties
- Format van user stories moet niet gevolgd worden: moet groot zijn, wie/wat/waarom wel belangrijk

Tasks

- Story kan onderverdeeld worden in uitvoerbare taken
- Geschreven door Dev team, voor Dev team
 - Mogen technische zaken bevatten
 - Hoeft geen directe business value te hebben
 - Moeten niet onafhankelijk zijn
 - Hoeveel niet vertical (end-to-end) te zijn

- Input van DevOps is belangrijk
 - Nodig systemen/hardware
 - Beperkingen naar security toe
 - Aparte technical tasks voor DevOps

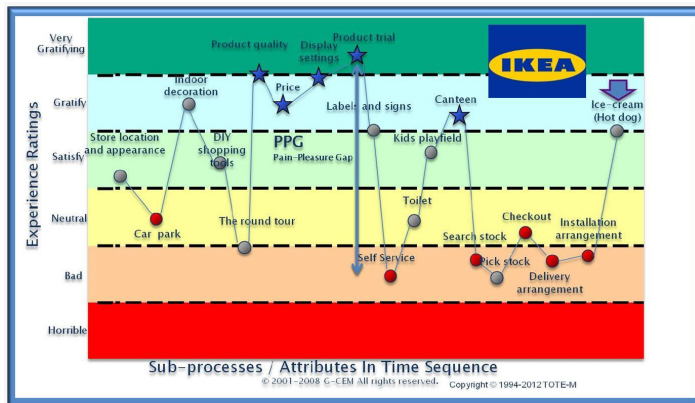
Belang story writing

- Business: create value
- Users: get a job done
- Delivery: make it feasible

→ Shared understanding

Customer journey

Voorbeeld customer journey: IKEA



- Visualiseert ervaring vd klant aangaande doelen/fases/activiteiten/contactpunten/emoties
- Doel: inzicht krijgen in gedrag vd klant
 - Toont mogelijks problemen
 - Identificeert opportuniteiten
 - Aligneert de organisatie
- Stappenplan
 - Creëren van persona's
 - Identificeren van verschillende fases, met doel en beschrijving vd activiteit
 - Identificeer contactpunten tussen klant en bedrijf
 - Beschrijf gevoel vd klant tijdens de actie
 - Identificeer de belangrijkste acties
 - Vind ideeën voor verbeteringen

Van customer journey naar story map

- Beslis wat onderzocht moet worden: bestaand/nieuw product/service, ...
- Bekijk customer journey maps, detecteer kansen/bedreigingen
- Beschrijf hypothese: ik denk..., het zou kunnen..., dat kan lijden tot...
- Identificeer de nodige capabilities

Story map



- Visuele oefening om shared understanding te creëren → business & IT
- Horizontaal: verhaal verteld
- Verticaal: activiteit verder uitgewerkt
- Als alle activiteiten aanwezig: prioriteiten toevoegen
- Start van een product backlog
- Voorbereiden - co-create - verify - prioritize
 - Prioriteren: belangrijk → niet alles kan even belangrijk zijn
 - Prioriteer story map: stel doel op Minimal Viable Product
 - Voeg perspectief van de klant toe: Minimal Lovable Product
 - Challenge the feasibility
 - Leg implementatie-scenario's vast: maak een roadmap

Prioriteren: MoSCoW

- Mo: Must have → essentiële elementen
- S: should have → belangrijke elementen
- Co: could have → leuke extraatjes
- W: won't have → voegen geen waarde toe

Product backlog

- Resultaat van story mapping
- Kan nog wijzigen doorheen project
 - prioriteiten veranderen
 - Must haves worden should/could of omgekeerd
 - Na iedere iteratie: feedback van klant en herbekijken

H04: valideren

Situering



Acceptance criteria

- GWT
- 3 amigo's
- Testplan

Valideren

- Begint met het bepalen van het succes
- 3 amigo's
 - Business: welke problemen willen we oplossen?
 - Development: hoe kunnen we het probleem oplossen?
 - Testing: lost deze oplossing het probleem kwalitatief op?
- Refinement sessies: user stories verduidelijken
 - Opsplitsen in kleinere stories
 - Acceptance criteria opstellen

Acceptance criteria

- Bepalen succes vd user story
 - Gebruikt door de testers om te valideren
- Gangbare manier om te definiëren: Given - When - Then
 - Behavior Driven (BDD)
 - User story als basis, meer details erbij voor GWT
 - Geeft aanzet naar functionele testen

Give/When/Then

Scenario: als een ingelogde user, wil ik kunnen inloggen op een website, zodat ik mijn persoonlijk profiel kan bekijken

GWT

- Given: ik ben een uitgelogde user, ik ben op de sign-in pagina
- When: als ik username & password invul met correcte credentials en ik klik op sign-in knop
- Then: systeem logt mij in

Testen

- Evolutie
 - Debuggen: onderdeel van programmeren
 - Zoeken naar fouten (los van programmeren)
 - Aantonen dat requirements voldaan zijn

Testen worden uitgebreider & gestructureerder

- Voldoen aan requirements
- Vinden van fouten
- Inzicht geven in kwaliteit
- Vertrouwen geven in kwaliteit van release
- Check op volledigheid & correctheid
- Advies geven over kwaliteit & risico's

Testen als vak

- Algemene kennis van informatietechnologie
- Kennis van infrastructuur, tools & ontwikkelomgeving
- Sociale vaardigheden
- Risk based testen
- Mensen, geld, tijd
- Verantwoorde wijze tests voorbereiden/uitvoeren
- Voldoende vertrouwen opbouwen om SW te accepteren

→ missing link tussen IT en business

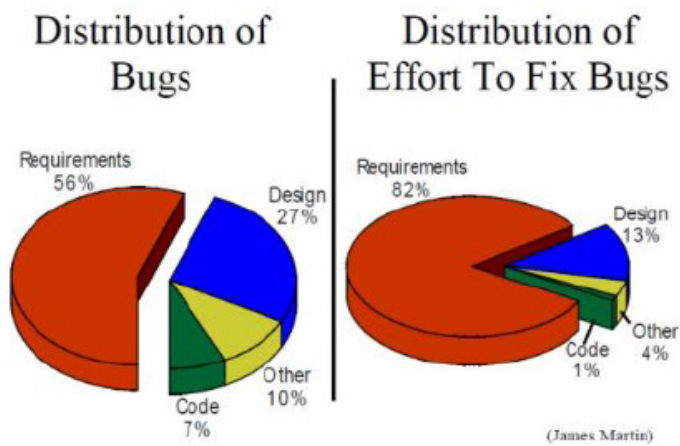
Waarom testen?

- Voor organisatie houdt elke wijziging een risico in
- Beperken van risico's: flexibiliteit verhogen
 - Softwareontwikkelp proces (Service Oriented Architecture)
 - Ontwikkelmethode die past bij de dynamiek van het bedrijf: agile development
 - Toepassen van controlemaatregelen: betrekken business bij opstellen/beoordelen systeemontwerp, uitvoeren van review/audits/...
 - Resultaatgedreven testen
 - Verificatie (het systeem goed bouwen)
 - Validatie (het goede systeem gebouwd is)
- Voorbeelden
 - Pentium-bug: FDIV bug (chip): niet genoeg precisie bij delen
 - Ariane 5: 2e vlucht: 2e trap niet losgekoppeld, oorzaak was SW-fout

Wanneer testen?

- In elke ontwikkelingsfase
- Inspanning: 30% tot 70% van het totaal
- Aandacht voor testen: afhankelijk van mogelijke gevolgen/risico's

Foutdistributie



Kost om defect te corrigeren

- Tijdens designfase: 1x
- Tijdens implementatie: 6.5x
- Tijdens testing: 15x
- Tijdens onderhoud: 100x

Schade aan klanten en gebruikers (dalende volgorde)

- Bedreigend voor veiligheid van menselijke wezens
- Aantasting ve essentiële organisatiefunctie
- Aantasting vd werking van firmware
- Aantasting ve essentiële organisatiefunctie, is een vervanging beschikbaar door alternatief systeem
- Aantasting vd goede werking ve software pakket voor business toepassingen
- Aantasting vd goede werking ve software pakket voor een private eindgebruiker
- Aantasting vd werking ve firmwaretoepassing maar zonder het volledige systeem te affecteren
- Brengt ongemakken voor de gebruiker mee maar verhindert hem niet om de systeemmogelijkheden te benutten

Schade aan de software-ontwikkelaar/organisatie

- Financieel verlies
 - Betaling schade voor fysieke blessures
 - Betaling schade aan organisaties voor de slechte werking van de softwareTerugbetaling van aankoopkosten aan klanten
 - Hoge onderhoudskosten voor het herstellen van falende systemen
- Niet-kwantitatieve schade
 - Negatieve impact op toekomstige verkoopcijfers
 - Substantiële terugval van de huidige verkoopcijfers

Toegevoegde waarde van testen

- Bijdrage aan het resultaat
- Oorzaak vd schade wegnemen
- Reductie van de impact
- Vertrouwen in het testobject
- Effectieve projectsturing

Binnen een SW-project: verschillende disciplines, belangen & aandachtspunten: projectmanager, programmeur, gebruiker, beheerder, business manager, tester

10 testprincipes

1. Focus op het resultaat
2. Bouw aan vertrouwen
3. Neem verantwoordelijkheid
4. Beheers het testvak
5. Sla bruggen
6. Test gefaseerd
7. Faciliteer de gehele IT-lifecycle
8. Geef overzicht en inzicht
9. Zorg voor herbruikbaarheid
10. Bedenk: testen is leuk!

1. Focus op het resultaat

- Focus op business resultaat
- Meer dan alleen bijdrage aan SW-kwaliteit
- Testen die bijdragen aan resultaten die de organisatie nastreeft
- Verificatie: aantonen dat systeem gebouwd is volgens systeemontwerp
- Validatie: inzicht geven in de mate waarin het systeem past binnen de bedrijfsprocessen

2. Bouw aan vertrouwen

- Vertrouwen opbouwen door intensief testen → minder kans op onopgemerkte fouten
- Probleem: beperkte resources (tijd, geld)
- Risicogebaseerd testen → belangrijke onderdelen grondiger testen
- Goede tester zoekt: fouten in andermans werk, goede elementen (positief kritisch)
- Goede tester betreft actief belanghebbenden bij het bepalen van risicogebieden, aandachtspunten → transparantie/volledigheid/expertise
- Verlies geen tijd en energie met het maskeren van fouten
- Beter op tijd een probleem
- Ervaring: bereidheid/begrip/aanvaarding van de consequenties indien dit tijdig gesignaleerd wordt

3. Neem verantwoordelijkheid

- Juiste testen voorbereiden
- Rekening houden met risico's
- Testdekking
- Overzicht testproject incl. controle
- Sterk betrokken bij IT-lifecycle
 - Slechte inventarisatie van de klantwens
 - Ondoordacht ontwerp
 - Onvoldoende kwaliteit programmeerwerk
 - Slecht configuratie- en versiebeheer

4. Beheers het testvak

- Testen is meer dan 'knoppen drukken'
- Goede kennis van systeemontwikkeling, testmethodieken
- Goede kennis van processen in organisatie
- Gesprekspartner voor de gebruikers/managers
- Bruggen slaan

5. Sla bruggen

- Veranderende requirements
- Inleving (belanghebbende)
 - De tester kan aan de ontwikkelaar uitleggen hoe de belanghebbenden naar het systeem kijken
 - De tester begrijpt de ontwikkelaar voldoende om samen met hem de bevindingen te analyseren

6. Test gefaseerd

- Gefaseerd testen resulteert in de beste SW-kwaliteit
- Duidelijke project aanpak, planning
- Logische volgorde
- Afdekking van de belangrijkste risico's
- Testproces
 - Strategie sluit aan op SW-ontwikkelp proces
 - KSF worden verdeeld over verschillende testsoorten (planning)
- Voordelen
 - Elke partij kan zijn voorbereidend werk doen
 - Entry- en exitcriteria(!)
 - Stap-voor-stap wordt duidelijk of het systeem aan de verwachtingen voldoet

7. Faciliteer de gehele IT-lifecycle

- Ontwerpfase
 - De producten toetsen op hun testbaarheid
 - Nagaan of het ontwerp aangeeft hoe het beoogde resultaat behaald gaat worden
- Ontwikkelfase
 - Hoe vroeger de fouten gedetecteerd worden hoe beter
 - Moduletesten, module-integratietesten
- Beheerfase
 - Delen van de testware
 - Goede overgang van projectfase naar beheerfase

8. Geef overzicht en inzicht

- Systeem = veel functies
- Per functie worden groot aantal testen uitgevoerd
- Testresultaten
 - Functietesten
 - Afdekken van risico's, aandachtspunten, kwaliteitsaspecten

9. Zorg voor herbruikbaarheid

- Tijdens het gebruik van de SW zullen nog fouten optreden
- Herbruikbare testset → automatiseren?
- Testtraject kost tijd en geld → herbruikbaarheid = kostenbesparen
- Risicoanalyse actualiseren

10. Bedenk: testen is leuk!

- Toegevoegde waarde binnen het SW-ontwikkelp proces
- Verantwoordelijkheid voor het in productie gaan van een systeem
- Flexibel en creatief zijn
- Specialisatie is mogelijk: testautomatisering, performance-testen en security-testen

Soorten testen

Moduletest (MT) & module-integratietesten (MIT).

- Geautomatiseerd
- Whitebox-testen
- Unit-testen, component-testen, programma-testen

Systeemtest (ST).

- Een heel systeem testen
- Blackbox-test
- Werkt het systeem conform het functionele ontwerp
- Bouwende partij

Functionele acceptatietest (FAT).

- Een heel systeem testen
- Opdrachtgever
- Verificatie a.d.h.v. systeemeisen en functioneel ontwerp

Gebruikersacceptatietesten (GAT).

- Validerende test
- Representatieve scenario's uit het dagelijkse werk van de gebruikers
- Gebruiksvriendelijkheid, bruikbaarheid, ...

Productieacceptatietesten (PAT).

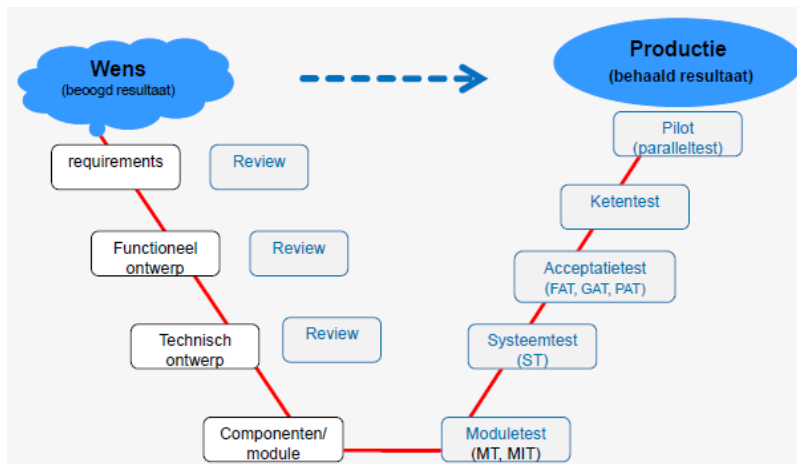
Ketentesten

- Samenwerking met de aanliggende systemen
- Integratietest

Pilot

- Parallel testen met bestaande systeem

Testen - vervolg



Verschillende testmethodologieën

- Agile testing
- Scripted testing
- Exploratory testing
- Rapid software testing

Afhankelijk van

- Project organisatie
- 'Application criticality'
- De testers

Testplan

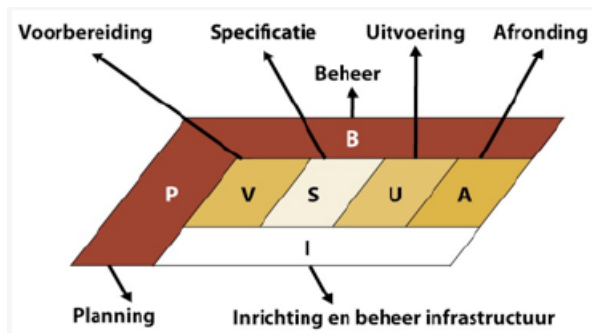
- Begin ve testtraject: testplan opstellen
- Doel vh testplan: aanpak vh testen binnen het programma/project/release/change te beschrijven
- Testplan opgesteld door de verantwoordelijke voor het testtraject → kan zijn de testmanager, de testcoördinator, de functioneel beheerder of ieder ander die de opdracht krijgt en aanvaard

Testplan

- Aanleiding → waarom
- Aanpak → hoe
- Benodigde mensen & middelen → wie en wat
- Benodigde planning → wat
- Op te leveren producten → wat

Onderwerpen die een rol spelen bij het opstellen van een testplan

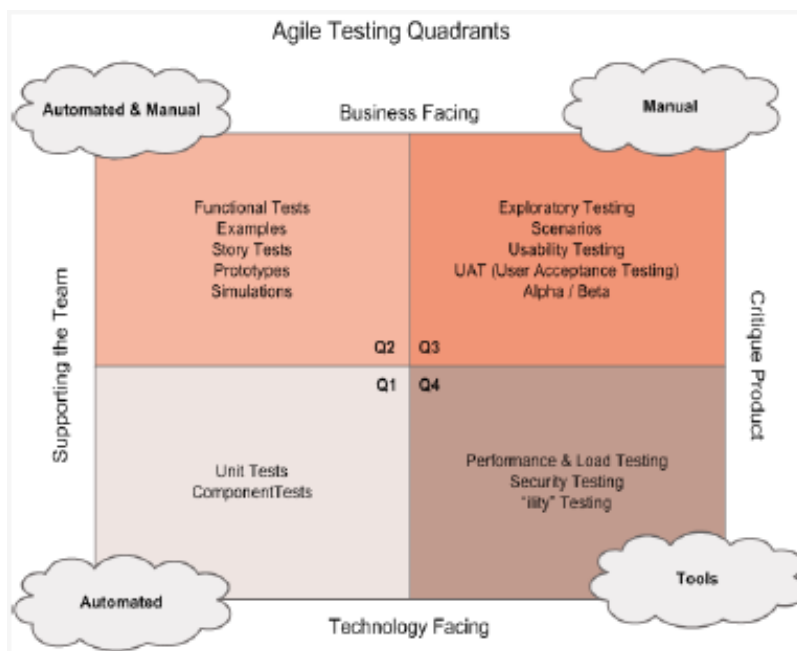
- Teststrategie
- Randvoorwaarden en uitgangspunten
- Entry en exit criteria
- Testsoorten entestvormen
- Testontwerpstechnieken



Wie voert de testen uit?

- Testers
- Programmeurs
- Business Analisten
- Ontwerpers (designers)
- Eindgebruikers
- Dedicated testing professionals
 - Opgeleid
 - Ervaren
 - 'Carriere'
 - Specialisatie
 - Security testers
 - Performance testing
 - Test automators
 - Embedded software testers
- Testing is een erkend beroep

Testautomatisering



Kwaliteitsborging

- Controle op basis van de opgeleverde tussenproducten of de vooropgestelde kwaliteit wordt bereikt
 - Vroegtijdig vaststellen van afwijkingen
 - Vroegtijdig herstellen
 - Kostenbesparend (Boehm)
- Hoe?
 - Collegiale reviews
 - Expert reviews
 - Inspecties
 - Walk-throughs
 - Monitoring (opdrachtgever) = audit

6 beoordelingsgrondslagen om een product te beoordelen



Toetstechnieken

- Collegiale review
- Expert view
- Structured walk through
- Inspectie

Collegiale review

- Collega's, die hetzelfde werk doen, beoordelen collega's
- Collega's zijn goed op de hoogte van omgeving, problematiek, technische achtergrond
- Minder vertrouwen in "vreemden" (auditor, inspecteur)
- Positief: collega's leren van elkaar

Risico's

- Te informele organisatie
 - Duidelijkheid geven over de review
 - Welk product, aspecten, hoeveel tijd, hoe bevindingen worden vastgelegd en hoe deze gecommuniceerd worden, aangeven wat prioritair is (review of eigen dagelijkse taak)
 - Bijzondere maatregelen
 - Training, uitvoeren van een pilot, ondersteuning bij uitvoering
 - Doel: collega's de meerwaarde van collegiale reviews laten onderkennen

- Geen eenduidige beoordelingsgrondslag
 - Vooraf bepalen welke aspecten in de review moeten belicht worden
 - Enkel inhoud?, voorgeschreven standaarden gehanteerd?, taalfouten corrigeren?, controle of vertaalslag correct gebeurd is? (Bv. FO naar TO), checklist?
 - Heldere beoordelingslag = verbetering van de kwaliteit
- De opsteller van het product en de reviewer dienen over voldoende communicatievaardigheden te beschikken
 - Collega's zien elkaar als concurrenten
 - "Afkragen" van het product wordt beschouwd als oncollegiaal gedrag
 - "Oncollegiaal gedrag moet voorkomen"
 - Maker van het product mag collegiale review niet zien als aanval op zijn werk
 - Doel van de review: het product verbeteren, niet het onderuit halen
 - Gedeelde verantwoordelijkheid
 - Product is niet van de maker alleen maar is een projectaangelegenheid
 - Kwaliteitsverbetering is een organisatiebelang
- Te vergevingsgezind
 - De reviewer meldt de geconstateerde afwijkingen niet

Conclusie

- Collega's beoordelen elkaars werk
- Zeer informeel
- Vooral gericht op inhoud
- Minder gericht naar vorm, toepassen van procedures
- Collega's leren van elkaar
- Voorwaarde: vertrouwen tussen de collega's

Expert review

- Expert die geen deel uitmaakt van het team
- Resultaten blijven intern
- Iets formeler van aard
- Gebruik van checklist, bevindingenlijst, rapport verhogen de kwaliteit van de review
- Minpunt: de expert kent de specifieke aspecten van de omgeving waarin de maker van het product werkt niet
- Pluspunt: de expert werkt onbevooroordeeld

Structured walkthrough

- De auteur leidt de groep door een document/product en licht achterliggende gedachte en keuzes toe
- Doel: consensus bereiken over de inhoud
- Auteur = actieve rol

Inspectie

6 fasen (Fagan)

- Planning
- Kick-off
- Voorbereiding
- Groepsmeetings
- Aanpassing product
- Afsluiting

Planning

- Toetsing entry criteria (=eisen waar een product aan dient te voldoen om geïnspecteerd te worden) → checklist
- Strategiebepaling: wie inspecteert welke aspecten?
- Verzamelen documenten: referentiekader opbouwen
- Benaderen inspecteurs

Kick-off

- Meeting: maker, inspecteurs, aspecten, belang

Voorbereiding

- Alle betrokkenen zoeken naar fouten

Groepsmeetings

- 3 meetings: Bevindingen-registratiemeeting, discussiemeeting, oorzaakanalyse
- Kwaliteitsmanager leidt vergadering

Aanpassing product

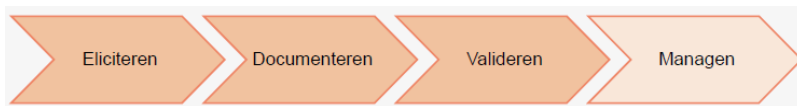
- Op basis van de bevindingen gebeuren aanpassingen
- Onafhankelijke functionaris (kwaliteitsmanager) checkt of alle bevindingen goed verwerkt zijn
- De maker blijft eindverantwoordelijke (hij kan sommige aanbevelingen bewust naast zich neerleggen)

Afsluiting

- Korte rapportage: auteur, inspecteurs, management

H05: managen

Situering



Product owner

- Heeft een visie voor het project
- Aanspreekpunt voor de stakeholders
- Beheert de Product Backlog
- Aanspreekpunt voor dev team

Recap - product backlog

- Resultaat van story mapping: geprioriteerde product backlog
- Kan nog wijzigen
 - Prioriteiten
 - Must haves → could/should haves
 - Na iedere iteratie/feedback: weten wat er gewenst is

Product backlog beheren

Stakeholders hebben veranderende eisen

- PO beslist of het in Product Backlog past
 - Value vs effort
 - Must have vs Could have
- Communicatie is key: met stakeholders, dev team, eindgebruikers

Hoe bepalen we value en effort

Estimating

- Value: MoSCoW
- Effort: man-uren, t-shirt sizing, story points

T-shirt sizing

- Relatieve manier om werk in te schatten
- Zeer ruw → eerst baseline

Size	Tijd	Size	Tijd
S	0-2 weken	L	4-8 weken
M	2-4 weken	XL	8-16 weken

Story_points

Moeten rekening houden met

- Hoeveelheid werk
- Complexiteit
- Risico
- Afhankelijkheid

Fibonacci wordt vaak gebruikt

- 1, 2, 3, 5, 8, 13 (20, 40, 100)

Refinement sessies

Scrumt stel voor regelmatig refinement sessies te houden

- PO bespreekt & schat stories in met dev team
- Hoogste prio's op Product backlog
- Zoveel mogelijk JIT werken
- Leren uit de feedback loop geeft betere inschattingen

Dev team kan effort inschatten aan de hand van Story Points (planning poker)

- PO legt story uit
- Team geeft individuele estimates
- Team overlegt en bespreekt de estimates
- Team komt tot een consensus over het aantal story points van de story

Sprint planning

- Product backlog in goede staat: prioriteiten en top stories
- Hoe bepalen wat volgende sprint haalt: velocity & yesterday's weather

Velocity

- Snelheid vh team: hoeveel werk kan er gerealiseerd worden binnen tijdslimiet
 - Bv. aantal story points binnen een sprint
- Tijdens sprint review sessie kan velocity berekend worden
 - Hoeveel story point zijn er opgeleverd in de afgelopen sprint?
 - Wat betekend opgeleverd? (Definition of Done)
 - Development klaar? Getest? Deployed? User Accepted?
- Anders voor beginnen team/project → fluctueert

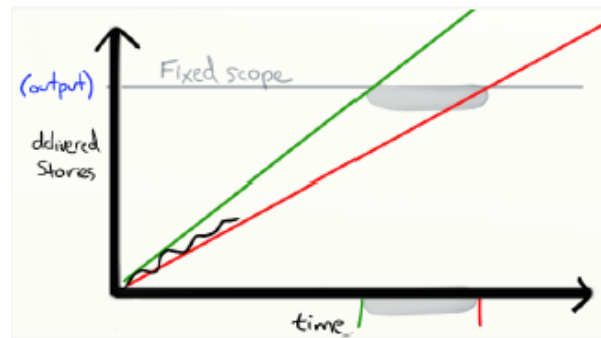
Yesterday's weather

- Kijken naar verleden om voorspelling te doen
 - Hoeveel Story Points zijn er vorige sprint opgeleverd?
 - Geeft een indicatie hoeveel werk er in de volgende sprint kan opgenomen worden
- Vaak wordt een gemiddelde genomen van x laatste sprints
 - Voorkomt uitschieters door onvoorziene omstandigheden (ziekte, verlof, ...)

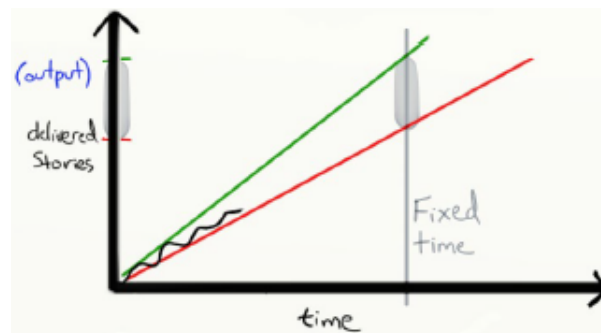
Forecasting

- Door velocity bij te houden → forecasting met 'Burn-up chart'

Wanneer zal 'x' klaar zijn?



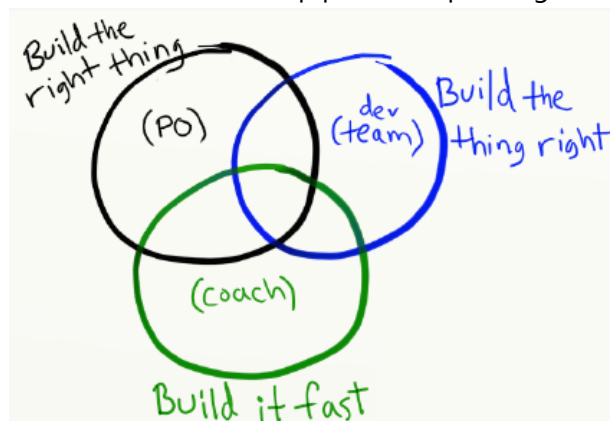
Wat zal er klaar zijn op moment 'y'?



Build the right thing

- Rol van PO → focus op juiste oplossing

Maar niet enkel focus op perfecte oplossing



Scrum master

Essentiële rol!

- Scrum process management
- Effectieve communicatie
- Team building & bescherming
- Progress tracking
- Quality check

Definition of done

Belangrijk onderdeel van goed werkend agile team: goed gedefinieerde DoD

- Scrum master kan dit faciliteren en afdwingen
- Opgesteld door Dev team, voor Dev team
- Bv. Getest, gedocumenteerd, deployed naar klant, PO approved, budget in orde

Agile principles

- Scrum master op zoek binnen andere agile methodologieën om een team te verbeteren
- Uit eXtremeProgramming (XP) wordt vaak volgende principes gehaald
 - (Automated) testing
 - Refactoring
 - Pair programming
 - Permanente integratie

Extreme programming

Automated testing

- Programma zonder geautomatiseerde test bestaat niet
- Allemaal op voorhand schrijven: onmogelijk → tijdsverlies

Hoe?

- Eenvoudig ontwerp
- Pair programming
- Vertrouwen
- Goed gevoel bij de gebruiker

Refactoring

- Permanent ontwerp en code wijzigen
- Tijdverlies, te weinig controle → crash

Hoe?

- "collective ownership"
- Eenvoudig ontwerp
- Standaard
- Pair programming/team
- Testen
- Permanente integratie

Pair programming

- Driver-navigator, piloot-copiloot
- Ervaring, one team
- Met 2 weet je meer dan alleen

Hoe?

- Gebruik standaarden
- Minder stress
- Samen testen schrijven
- Samen eenvoudig ontwerp uitwerken
- Samen implementeren

Permanente integratie

- Code wordt geïntegreerd en getest na enkele uren

Hoe?

- Zeer snel testen
- Voortdurende refactoring
- Je werkt met 2

Continuous integration: evt. Jenkins

H06: processen

Situering

Ieder systeem ontwikkeld om bepaalde reden

- Beter procesbeheer
- Lagere kosten
- Beter benutten van commerciële mogelijkheden
- Verhogen vd servicegraad

→ Vooraf eisen helder en 1duidig formuleren = business case

Business process management vs Business analyse

BPM

- Gericht op bedrijfsorganisatie
- Meestal voortraject van IT-project

Business analyse

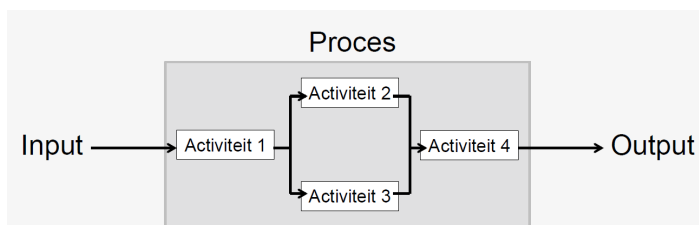
- Vertrekt van resultaat BPM-oefening
- Gericht op IT-project
- Tijdens business analyse blijkt vaak nood aan procesoptimalisatie

1. Definitie van een proces

Definitie ISO 9000:2005: "een geheel van samenhangende/elkaar beïnvloedende activiteiten die input omzetten in output"

Definitie

- Transformatie van input naar output
- Creëren van waarde voor afnemer of klant (=belanghebbenden)
- Realiseren van doelen: bepaald door verwachtingen vd afnemers of belanghebbenden



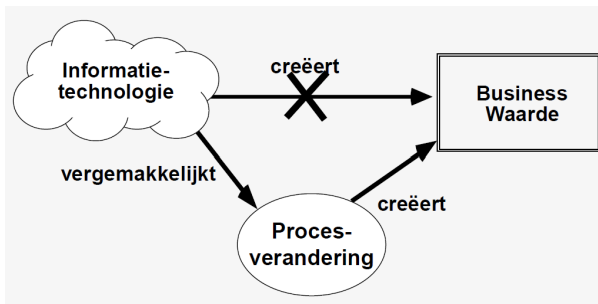
Opmerkingen

- Input in proces: output van ander proces
- Processen in het algemeen gepland/uitgevoerd onder beheerste omstandigheden

Voorbeeld ve proces: afhandeling van IT-incident (doorheen de hele keten)

2. BPM (waarom - hoe)

Waarom



Hoe

- Continuous Process Improvement (CPI)
 - Stelt huidige processtructuur niet in vraag
 - Identificeert problemen en lost ze 1 voor 1 op
- Business Process Re-engineering (BPR)
 - Stelt fundamentele veronderstellingen/principes vd huidige processtructuur in vraag
 - Gericht op het realiseren van een doorbraak

Case study Ford (Hammer 1990).

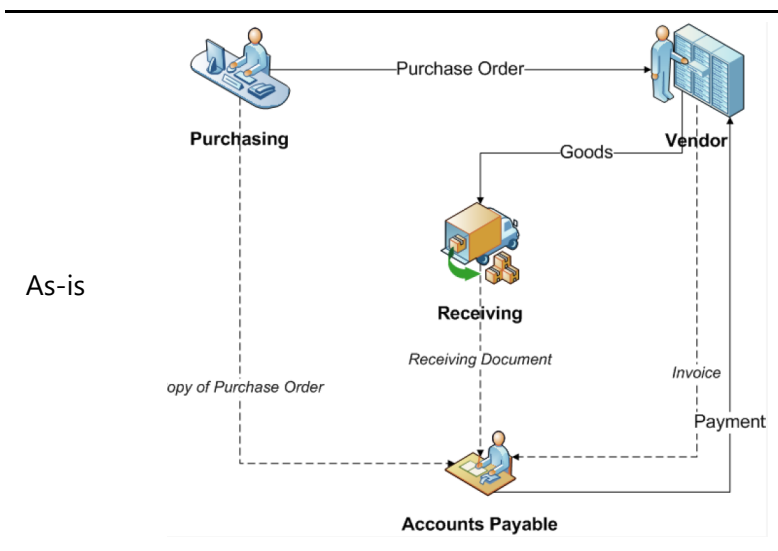
Ford moest zijn aankoopproces hierzien om

- Het goedkoper te doen → kostenreductie
- Het sneller te doen → kortere doorlooptijden
- Het beter te doen → minder fouten

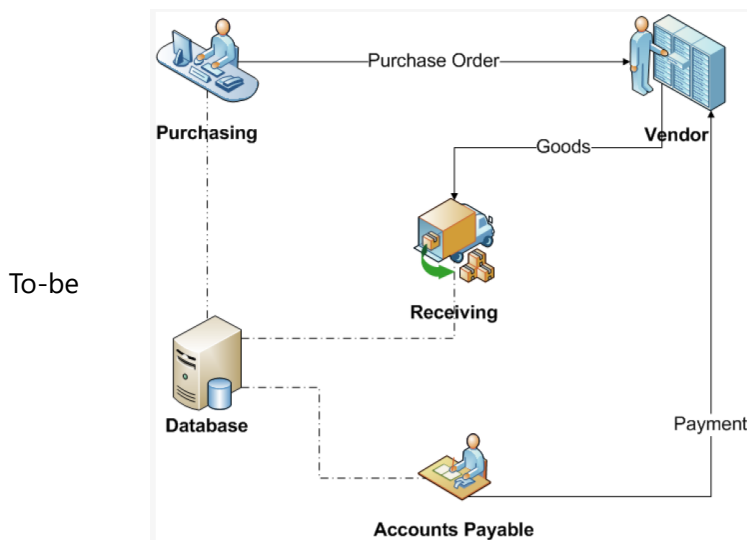
Situatie as-is

- In afdeling 'Leveranciersfacturen' > 500 mensen
- Doorlooptijden voor verwerken van uitgaande bestellingen: grootte-orde van werken
- Automatisatie: verbetering van 20%

Situatie Voorstellingen



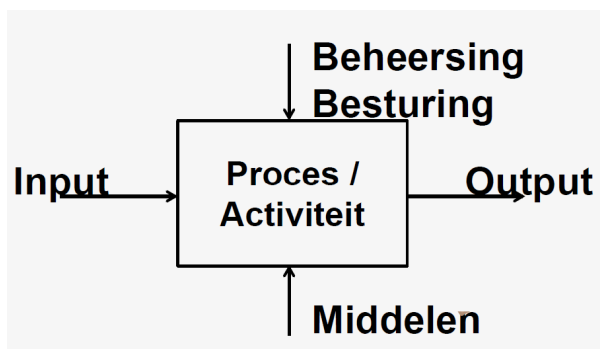
Situatie Voorstellingen



Resultaat

- 75% minder personeel
- Materiaalcontrole is eenvoudiger
- Financiële controle is accurater
- Snellere afhandeling van aankopen
- Minder laattijdige betalingen

3. Verklaring van enkele begrippen



Input: iets dat getransformeerd/verbruikt/verwerkt wordt (bv klantenvraag)

Output: iets dat geproduceerd wordt (bv attest, antwoord)

Proces/activiteit: hoe/wanneer een proces/activiteit plaatsvindt, wordt niet verwerkt/verbruikt (bv instructie, richtlijn)

Middelen: personen/systemen/tools/uitrusting/activa, niet verwerkt/verbruikt, wel gebruikt (bv kopieerapparaat, IT-pakket)

4. End-to-end (van uiteinde tot uiteinde)

Proces

- Begint bij behoefte van externe klant/afnemer → triggert proces
- Eindigt bij zelfde klant/afnemer → trigger beantwoord

Proces vs functie

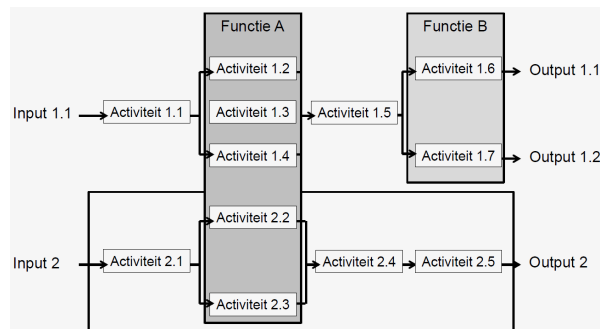
Proces

- Verzameling activiteiten
- Gericht op het bereiken van 1 of meer outputs

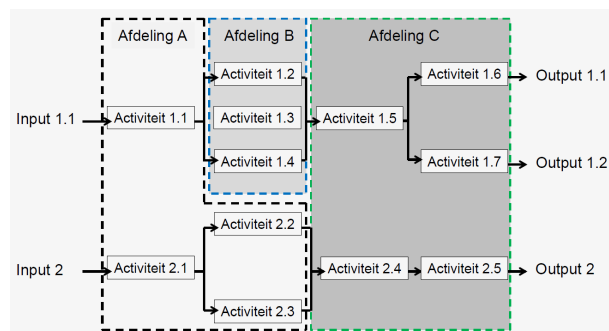
Functie

- Verzameling van activiteiten
- Gegroepeerd volgens competentie

Proces vs functie



Proces vs afdeling



Belang van processen

Processen

- Hart ve organisatie
- Lopen vaak fout bij transfer tussen andere persoon/afdeling/instantie

2 belangrijke dimensies in organisatie

- Strategische dimensie: juiste dingen doen → effectiviteit
 - Inspanningen ve organisatie moeten gericht zijn op verbeteren vh vermogen om waarde te creëren
 - Kosten vs baten vh resultaat afwegen
 - 1e stap BPM: identificeren vd 'waarde creërende' stromen
- Operationele dimensie: dingen goed doen → efficiëntie

Belang - recap

- Creëren van waarde: gebeurt via processen
- Niet alle processen gericht op waarde creëren
 - Niet alle processen zijn evenwaardig
 - Vaak stappen zonder toegevoegde waarde
 - Identificeren van deze processen: 1e stap
 - Primair, ondersteunend, sturend
 - Kritische succesfactoren
 - Essentiële processen?
 - Hoe performant zijn ze?
- Waarde creërende processen
 - Meestal primaire processen
 - Gelinkt aan kerncompetenties vd organisatie
 - Van kritisch belang
- Toegevoegde waarde
 - Beantwoorden aan behoefte van belanghebbenden
 - Moeten een doel hebben
 - Staan centraal en moeten meerwaarde bieden
 - Zonder toegevoegde waarde: geen proces → afvoeren

Procesmodel

- Ordenen van sleutelprocessen
- Visuele voorstelling op hoogste/globale niveau vd organisatie
- 3 clusters van processen
 - Primaire
 - Ondersteunende
 - Sturende



Sturende processen

- 'management processen'
- Richting geven & doelen stellen
- Evaluatie en controle
- Gedrag vd organisatie
- Bv. project management proces

Primaire processen

- 'core business processen'
- Productontwikkeling, productie, dagelijkse operaties, sales & marketing
- Bv. van order-intake tot facturatie, van product-idee tot product-lancering

Ondersteunende processen

- 'support business processen'
- ICT, boekhouding, HR-afdeling (afhankelijk van core vh bedrijf)
- Bv. aanwervend ve nieuw personeelslid, goedkeuring inkomende facturen

Problemen bij processen

- Functioneel georganiseerde organisaties
 - Functie/afdeling → verticaal
 - Proces → horizontaal
- Dalende doeltreffendheid door groei
 - Groei → specialisatie: efficiënter op bepaald domein, maar verlies einddoel/klant uit het oog
- Gewoontevorming

Processen in verschillende organisatietypes

Verticale functionele 'kachelpijpen'

- Elke kachelpijp: specifieke functie/dienst (IT, marketing, financiën, HR, R&D)
- In hiërarchische, gecentraliseerde organisaties
- Expertise wordt gedeeld over het bedrijf
- Duidelijke carrière-paden en opleidingsprogramma's
- Voor elke functie bestaan backups
- Managers zijn vertrouwd met het werk van hun ondergeschikten
- Standaarden kunnen gemakkelijk gehanteerd worden
- Nadelen
 - Eeneenheidop zich (eigen bureaucratie)
 - Focus op de individuele dienstverlening (Bv. IT) i.p.v. op de dienstverlening aan de klant
 - Processen zijn gericht op efficiëntie (dingen juist doen), niet op effectiviteit (de juiste dingen doen voor de organisatie)
 - Communicatieproblemen (eigen jargon)
 - Business prioriteiten kunnen afwijken van de functionele prioriteiten
 - Opgeleverde projecten voldoen niet aan de noden vd business units

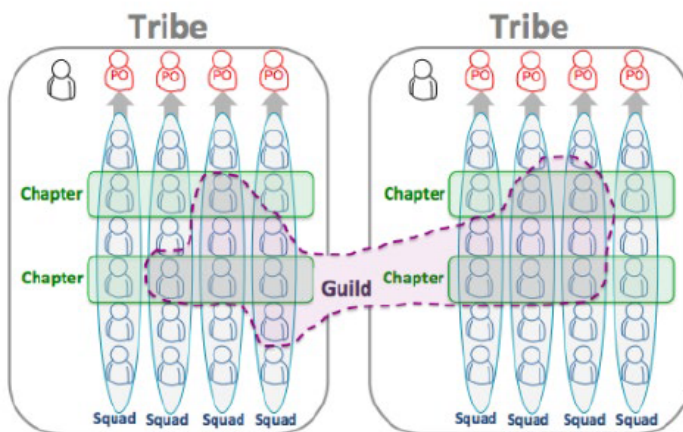
Horizontale process-tunnels

- In gedecentraliseerde organisaties
- Units gecreëerd om te focussen op bepaald business-domein
- Prioriteiten zijn gebaseerd op de noodzaak van producten & processen
- De communicatie tussen de diensten is veel beter
- Het personeel is op de hoogte van diverse aspecten vd producten & processen

- Nadelen
 - De perceptie: de opwaartse carrièrepaden beperkt
 - Jobevaluaties gebeuren vaak door leidinggevenden die de job zelf niet kennen
 - Vaak weinig backup-personeel
 - Weinig synergie in het bedrag
 - Iedere business unit werkt op zijn eigen manier, wat kan leiden tot redundantie en inefficiëntie

Nieuwe trend

Agile werken → Spotify as a Framework



5. Starten met procesverbetering

Kritische-/sleutel-processen

- Staan in functie van de strategie van een organisatie
- Dragen bij tot het voldoen aan de behoeften van de belanghebbenden
- Zijn de sleutel tot het succes of continuïteit van de organisatie
- Worden bepaald op basis van kritische succesfactoren

→ Vertrekpunt: SWOT analyse

Sterktes/zwaktes

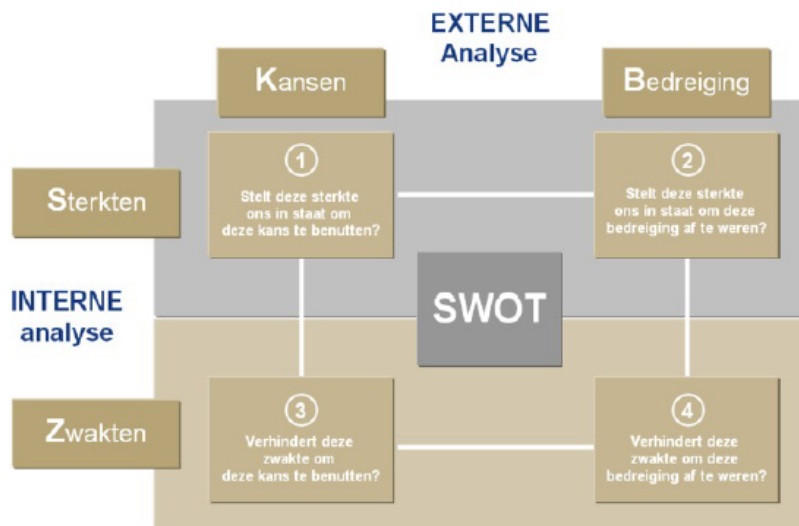
Sterktes: waarmee onderscheiden we ons

- Kritische succesfactoren
- Kerncompetenties
- Merk, R&D, productie, produktontwikkeling, kost- & prijsstructuur, verkoop- & distributiekkanalen
- Technologische capaciteiten

Zwaktes: observatie van bestaande business- en IT-problemen

- Verkeerde skills van mensen
- Omslachtige procedures
- Logge, hiërarchische structuur

Starten met processen



Kritische/sleutelprocessen voorbeelden

- Verbeteren van de kostenstructuur
- Groei
- Versterken van imago
- Verhogen van de klanttevredenheid
- Verhogen van de marketinginspanningen
- Verhogen van de competentie van het personeel
- Verbeteren van de interne communicatie

H07: Business process modeling - deel 1

Doel van modelleren

Doel: reduceren van complexiteit, vereenvoudigen vd werkelijkheid

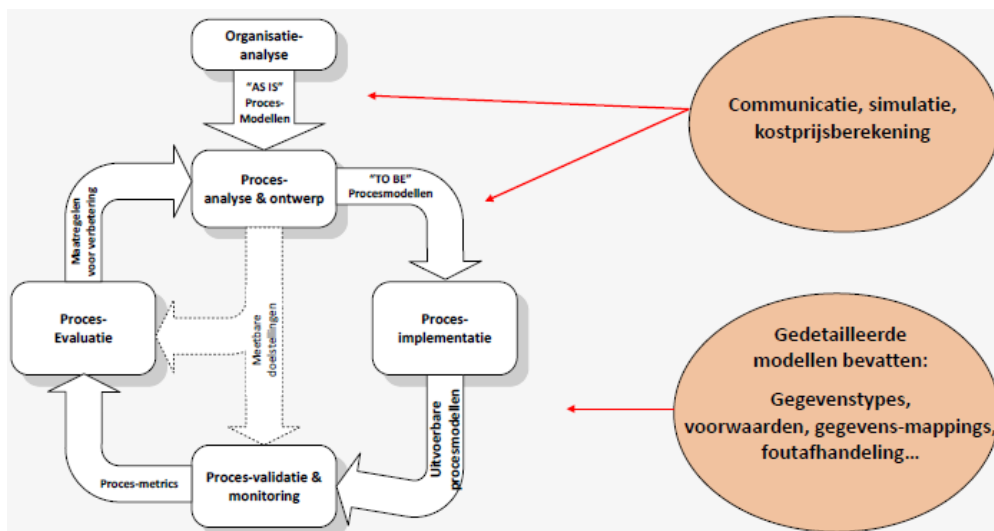
Voorwaarden

- Goed model bevat alleen die aspecten waar je grip op probeert te krijgen
- Modelleren is kiezen tussen 2 uitersten
 - Te weinig aspecten: model is te simpel, weinig zinnige conclusies
 - Te veel aspecten: model is te complex, even ingewikkeld als werkelijkheid
- Goed model bevat een boodschap → communicatiemiddel

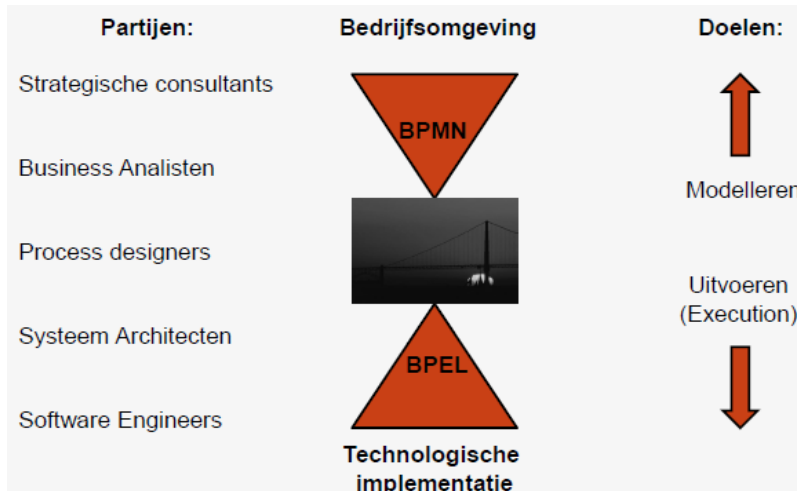
Wat

- Moeten 1 boodschap bevatten
- Zijn een communicatiemiddel
- Geven antwoord op concrete vragen
- Boodschap moet duidelijk zijn in < 10 minuten
- Modelleren is meer dan tekenen van diagram
- Alle modellen zijn onvolledig, sommige nuttig
- Er bestaan meerdere modellen van 1 proces

Doel



Notatietechnieken voor BPM



Oorsprong

BPEL

- Business Process Management Institute (BPMI)
 - Onderdeel van Object Management Group (OMG)
 - Ontwikkelt BPML (XML Process execution taal)
 - Realiseert behoefte voor een grafische weergave
- BPML later vervangen door BPEL
 - Business Process Execution Language
 - BPEL als doel uitvoeringstaal
- 2001: Notation Working Group opgericht

BPMN

- BPMN 1.0 (23 events): 2004, 1.1 (26 events) in 2008, 1.2 in 2009
- BPMN 2.0 (63 events): 2011

Business Process modeling Notation (BPMN) - tools

- Bizagi Process Modeller
- Aris express
- Visual paradigm
- Microsoft visio
- Draw.io

BPMN in vogelperspectief

4 types elementen

- Event
- Task
- Flow
- Gateway

Gateways

Exclusive decision/merge

- Delen ve proces kunnen 2 of meer alternatieve paden volgen
- 1 vd paden wordt effectief genomen
- Ruit met optioneel 'X'

Parallel fork/join

- Parallele flows creëren (fork) en synchroniseren (join)
- Ruit met '+'

Organisatie-elementen in procesmodellen

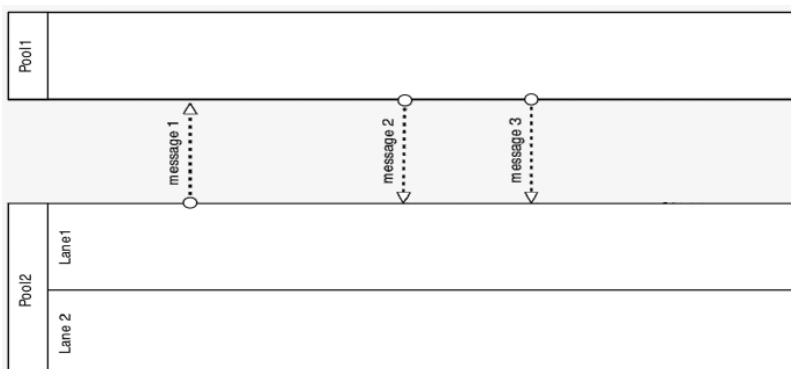
2 basis-abstracties

- Resource
 - Menselijke actor/toestel nodig om een activiteit uit te voeren
- Resource-klasse
 - Verzameling van resources met dezelfde eigenschappen
 - Een rol (competentie, kwalificatie)
 - = classificatie op basis van wat een resource kan/moet doen
 - Een groep (afdeling, team, kantoor, organisatie-eenheid)
 - = classificatie op basis van organisatie-structuur
 - Bv. administratief bediende, manager, ...

Resource modelling in BPMN

- Zwembaden
 - Onafhankelijke organisatie-eenheden
 - Bv. klant, leverancier, overheid, ...
- Zwembanen
 - Resource-klassen in zelfde organisatie
 - Bv. Verkoop, marketing, productie, HR, ... / Boekhouder, manager, ingenieur, ...

Zwembaden en -banen - notatie



Voorstellen v/e proces via BPMN-zwembanendiagram

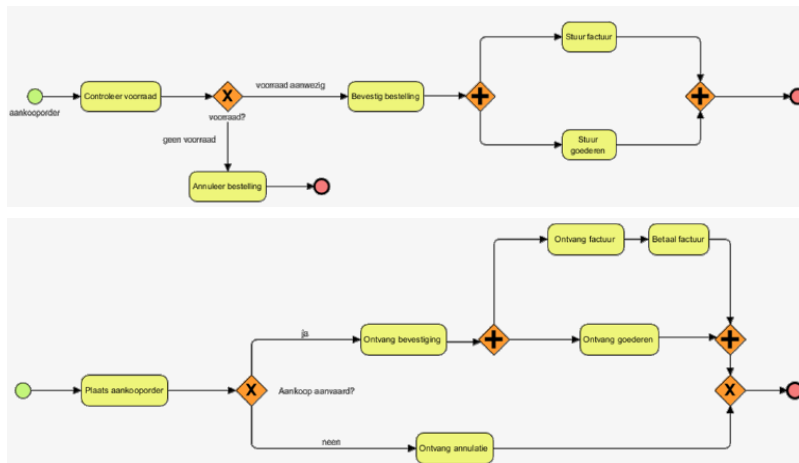
Beperkingen	Voordelen
Vele poorten verminderen leesbaarheid	Rollen duidelijk weergegeven
Geen andere objecten dan rollen, activiteiten, outputs	Activiteiten en volgorde duidelijk weergegeven
	Inputs & outputs tussen rollen zijn zichtbaar

BPMN informatie-artefacten

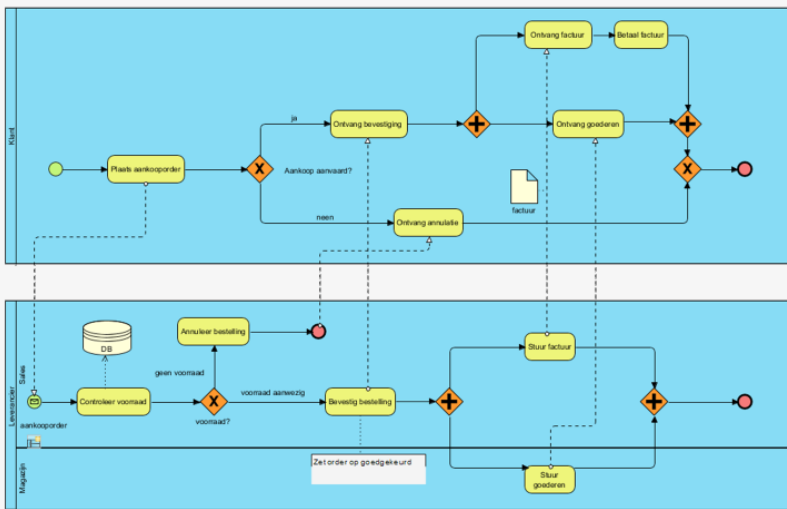
- Data-objecten
 - Mechanismen om aan te geven hoe activiteiten data gebruiken/produceren
 - Stellen input/output van activiteit voor
- Data-stores
 - Containers van data-objecten die moeten gepersisteerd worden na levensduur van proces
- Associaties
 - Link leggen tussen artefacten zoals data-objecten en data-stores met flow-objecten
- Text annotations (=commentaren)
 - Extra info voorzien voor de lezer

Praktisch voorbeeld

Komende van:



Naar (met zwembaden en informatie-artefacten):



BPMN poorten

Exclusief (XOR) → 'X' in ruit

- Exclusieve beslissing: 1 van de paden wordt genomen
- Exclusieve merge: ga verder als 1 tak beëindigd is

Parallel (AND) → '+' in ruit

- Parallele splitsing: alle paden worden genomen
- Parallele join: ga verder als alle takken beëindigd zijn

Inclusief (OR) → 'O' in ruit

- Inclusieve beslissing: kies 1 of meer takken
- Inclusieve merge: ga verder als alle actieve inkomende takken beëindigd zijn

Activiteiten

Kenmerken

- Zekere duurtijd
- Verbruikt middelen
- Uitgevoerd door een rol
- Altijd getriggerd door
 - Vorige activiteitenblok
 - Evenement of gebeurtenis: boodschap, tijd, regel
- Genereert altijd minstens 1 (intermediaire) output
- Uitgevoerd op 1 of meer fysieke plaatsen

Formulering

- Goed: 'verzenden vergunning'
- Fout: 'beheren vergunning', 'verbeteren vergunning', 'verzenden formulier F45', 'verzenden'

Veel gebruikte werkwoorden

Afwerken	Archiveren	Beoordelen	Bepalen	Berekenen
Beslissen	Bijwerken	Controleren	Creëren	Evalueren
Goedkeuren	Informereren	Invullen	Lanceren	Plannen
Produceren	Registreren	Toetsen	Uitvoeren	Valideren
Veranderen	Vergelijken	Vergelijken	Vorbereiden	Wijzigen

Verbindingselementen

Sequentiële stroom

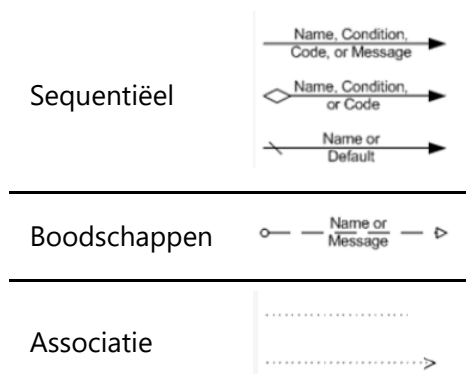
- Enkel binnen 1 zwembad
- Niet tussen zwembaden

Boodschappenstroom

- Enkel tussen verschillende zwembaden

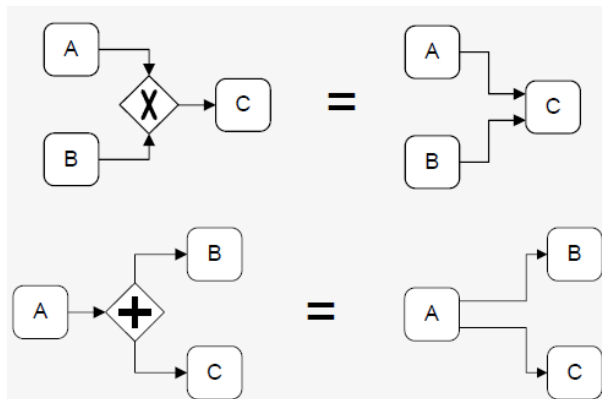
Associatie

- Verbindt informatie en artefacten met stroomelementen



H07: Business process modeling - deel 2

Expliciete vs impliciete controle-flow (semantiek)



Expliciet (linkse manier van werken): open-/sluit-poorten neerschrijven

Impliciet (rechtse manier van werken): open-/sluit-poorten niet neerschrijven, duidelijk uit context wat het is

Default flows









- Tak die gekozen wordt bij een (X)OR-splitsing als alle andere condities niet waar zijn
- Vermijden deadlocks
- Niet noodzakelijk de meest gebruikte keuze
- Niet verplicht te gebruiken

Start/end events

Start events

Geen		De modelleerder geeft het type startevenement niet weer. Wordt ook gebruikt voor een subprocess dat start wanneer de stroom getriggerd wordt door het moederproces.
Boodschap		Een boodschap (bericht, mail, brief, melding,...) komt van een participant en triggert de start van het proces.
Tijd		Een specifieke tijd/datum of specifieke cyclus (elke maandag om 9.00u) die de start van het proces triggert.
Voorwaardelijk		Wanneer een voorwaarde (vb. de winst is lager dan x €, indicator hoger dan y) WAAR komt.
Signaal		Een signaal komt aan dat uitgezonden werd door een ander proces. NB: een signaal heeft geen specifiek vooropgestelde bestemming. Een boodschap heeft dit wel.
Meervoudig		Er zijn verschillende manieren om het proces te triggeren. Een van de types is voldoende om het proces te doen starten.

End events

Geen		De modelleerder geeft het type eindevenement niet weer. Wordt ook gebruikt om het einde van een subproces weer te geven waardoor de stroom naar het moederproces gaat.
Boodschap		Een boodschap (bericht, mail, brief, melding,...) wordt naar een participant gezonden bij het einde van het proces.
Fout (Error)		Een fout wordt gegenereerd en opgepikt door een fout-evenement dat aan de rand van een activiteit weergegeven wordt.
Annulering (Cancel)		Alleen te gebruiken bij transacties (die niet succesvol uitgevoerd zijn)
Compensatie		Geeft aan dat een compensatie nodig is voor een aangeduide activiteit. Om gecompenseerd te worden moet een activiteit aan de rand van het object een compensatie evenement hebben.
Signaal		Een signaal wordt uitgezonden wanneer het einde van het proces bereikt is. NB: dit signaal kan uitgezonden worden doorheen de procesniveaus of de zwembaden
Einde (Terminate)		Geeft aan dat alle activiteiten van het proces onmiddellijk moeten beëindigd worden (zonder compensatie of evenementafhandeling)
Meervoudig		Er zijn verschillende manieren om het proces te beëindigen.

Intermediate events

Event dat plaatsvindt ergens tussen begin en einde van proces

- Sturen/ontvangen berichten
- Toont vertraging aan
- Onderbreekt normale flow voor een uitzondering
- Toont extra werk voor een compensatie

Geen		De modelleerder geeft het type event niet weer. Wordt gebruikt om aan te tonen dat er geen specifieke trigger is. Toont een statusverandering aan in het proces.
Timer		Een vertragingindicator, b.v. 5 dagen
Boodschap (Catching)		Dit event geeft weer dat het proces wacht op een bericht alvorens verder te gaan. Zolang het bericht niet ontvangen wordt staat het proces 'on-hold'. Boodschappen komen steeds uit een ander zwembad/proces.
Boodschap (Throwing)		Het proces verstuurt een bericht naar een ander zwembad/proces
Signaal	<div>Throw</div>  <div>Catch</div> 	<p>Wordt gebruikt voor het versturen/ontvangen van signalen uit andere zwembaden/processen.</p> <p>Het versturen van een signaal is een 'broadcast' naar allen die geïnteresseerd zijn.</p>

'Catching' vs 'Throwing'

Catching: boodschappenpijlen komen toe naar het evenement

Throwing: boodschappenpijlen vertrekken van het evenement

Uitzonderingsstromen

Uitzonderingsevenement

- Gebeurtenis tijdens de uitvoering van het proces
- De normale stroom van het proces wordt afgeleid buiten de normale stroom
- Kunnen gegenereerd worden door een time-out, fout, boodschap, ...

Uitzonderingsstroom

- Sequentiële stroom die start met uitzonderingsevenement aan de grens van een activiteit
- Enkel doorlopen wanneer uitzondering zich voordoet
- Normale stroom kan onderbroken worden
 - Interrupting vs non-interrupting

Soorten activiteiten



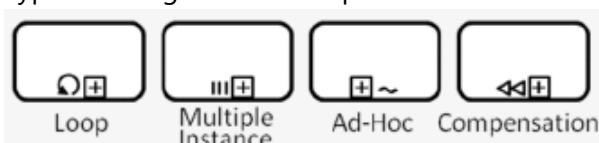
Nummers uit afbeelding

1. BPEL-gestuurd systeem
2. ERP-systeem
3. Indien de taak meer bevat dan versturen/ontvangen, catch/throw-event gebruiken

Proces-decompositie

- Activiteit ontleden in subprocess
- Gebruik dit om
 - Grote modellen op te delen in kleine, duidelijke onderdelen
 - Onderdelen vh procesmodel te identificeren die moeten worden
 - Herhaald
 - Meermaals in parallel
 - Afgebroken
 - Gecompenseerd

Types samengevouwen subprocessen



- Lus (loop)-subproces
- Meerdere verzoeken (multi-instance)-subproces

- Ad-hoc proces
= groep van activiteiten die geen voorgedefinieerde sequentiële relaties hebben
- Compensatie-subproces
= groep van activiteiten die uitgevoerd worden bij het 'terugdraaien' van een transactie om activiteiten te compenseren die uitgevoerd werden tijdens de normale stroom van een proces

Wanneer

- Als model te groot wordt: moeilijk te begrijpen, verhoogde kans op fouten
- Vuistregel: max 30 flow-objecten per model

Proces hergebruiken

Subproces ingebed in ouder-proces

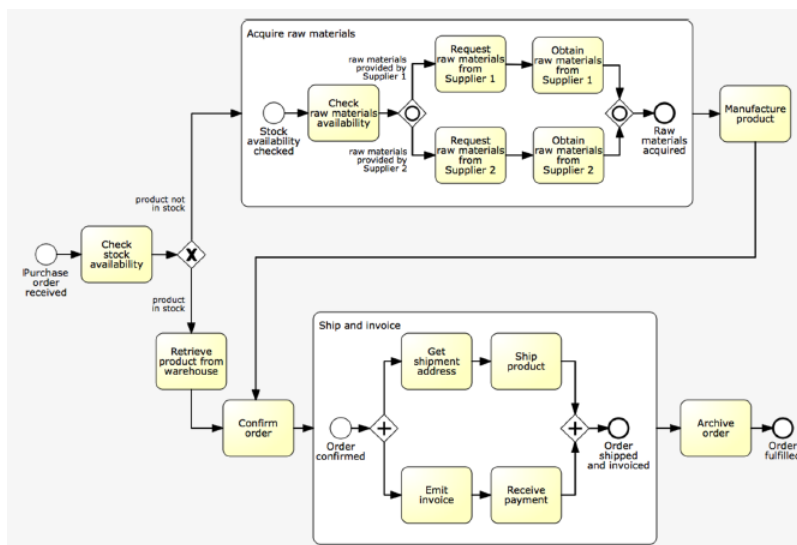
- Opgeslagen in zelfde bestand

Hergebruik maximaliseren

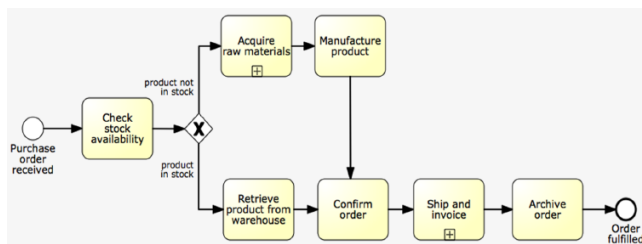
- Subproces 'uit bestand te trekken'
- Subproces is dan globaal en kan aangeroepen worden via een call-activiteit (dikkere rand)

Praktisch voorbeeld

Van:



Naar:

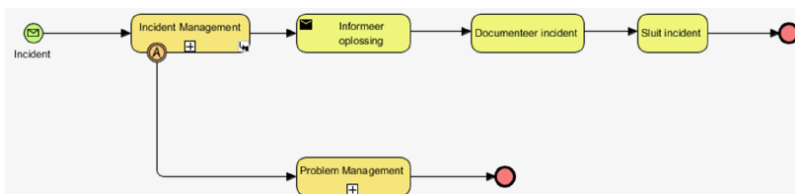


Syntax-regels voor subprocessen

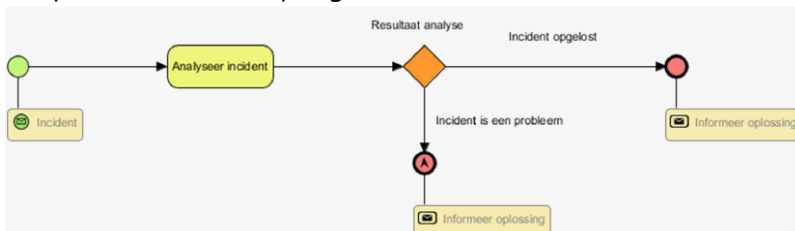
- Start met minstens één start-event
 - Indien meerdere, dan zal de eerste die voorkomt, het subprocess triggeren
- Eindig met minstens één eind-event
 - Het subprocess zal beëindigd zijn als alle eind events bereikt zijn. Hierdoor kan een (X)OR-splitsing na het subprocess nodig zijn om aan te geven welk(e) event(s) opgetreden is(zijn)
- Sequence flows (pijlen) kunnen de grenzen van een subprocess niet overschrijden
 - Gebruik start/end events
- Message flows kunnen wel de grenzen van een subprocess overschrijden
 - Om aan te geven dat messages in of uit een subprocess kunnen gaan.
 - Signalen/escalaties/errors kunnen verstuurd worden

Hoofdproces met uitzonderingsstroom

- Proces wordt onderbroken door een escalatie en triggert Problem Management
- Normale flow wordt nooit afgewerkt

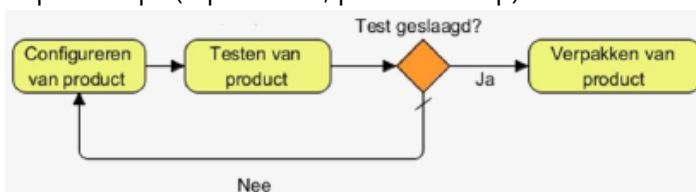


Subproces met interrupting escalatie:

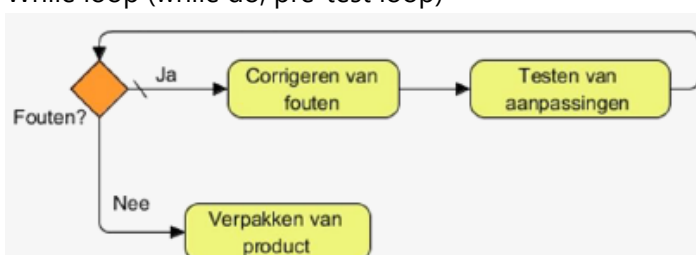


Standaard lussen (loops)

Repeat loops (repeat until, post-test loop)

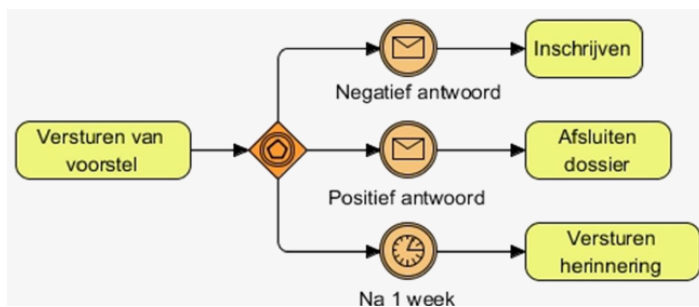


While loop (while do, pre-test loop)



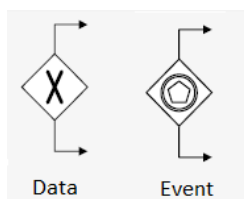
Event-driven

Ports



Gateways

- Bij een data-driven XOR-splitsing, wordt één tak gekozen, op basis van beschikbare gegevens
 - De keuze kan onmiddellijk gemaakt worden als de "estafettestok" aankomt bij de poort
- Soms echter moet de keuze uitgesteld worden tot er zich een gebeurtenis voordoet
 - De keuze is gebaseerd op een race tussen events
- Daarom maakt BPMN een onderscheid tussen data-gedreven en event-gedreven XOR-poorten



Hoofdproces met niet-onderbrekende uitzonderingsstroom

- 2e uitzondering kan geactiveerd worden maar onderbreekt de normale flow niet
- Subproces blijft actief

Subproces met 2e escalatie

- Eerstkomend event maakt ander pad onmogelijk

Artefacten

Data-objecten

- Geen stroomobjecten (niet verbonden doorheen de sequentiële stroom)
- Voorzien in informatie over de manier waarop documenten, data en andere objecten gebruikt en aangepast worden binnen een proces

Groepen

- Organiseren op een visuele wijze activiteiten in een cluster die niet anders kan voorgesteld worden in het diagram
- Niet gebonden aan beperkingen van zwembaden of zwembanen

Data-objecten

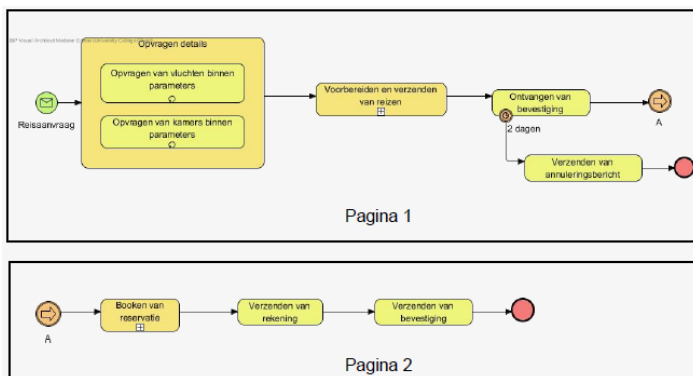


Groepen



Off-page connectoren

- 'bol' met pijltje
- Stop op ene pagina en gaat verder op de volgende



Vaak voorkomende fouten

- Ontbrekende labels bij (X)OR-splitsingen, message flows, start/end events
- 2 acties in 1 taak → 2 taken
- Taken van een te laag niveau → samenvoegen indien zelfde resource
- Normale sequence flow over zwembad-grenzen → gebruik messages

Stappenplan BPMN-diagram

- Bepalen van het doel of de finaliteit van het BPMN-diagram
- Bepalen van de context van het proces
- Bepalen van het aantal zwembaden en zwembanen
- Bepalen van het begin- en eindevenement van elk zwembad
- Bepalen van de verschillende activiteiten, poorten, tussentijdse evenementen en sequentiële stromen binnen elk zwembad
- Bepalen van de verschillende boodschappenstromen tussen de zwembaden
- Vervolledigen van het diagram met artefacten waar nodig

Tips modelleren in BPMN

- Benoem alle objecten in het diagram (behalve de sequentiële stromen)
- Valideer het diagram ten opzichte van de BPMN-regels
- Maak de modellen hiërarchisch
- Start de benoeming van elke activiteit met een actief werkwoord (infinities)
- Vermijd activiteiten als 'Versturen naar...' of 'Ontvangen van...' wanneer het een informatieoverdracht betreft binnen 1 zwembad. Deze activiteiten worden vervangen door een sequentiële stroom
- Communicatie aan een manager zonder dat het bijbehorende werk (Bv. dossier) = de activiteit "Informereren van..." in de zwembaan van de zender zonder sequentiële stroom naar de manager
- Indien meerdere eindevenementen zijn dan is het niet nodig om deze te laten samenkomen naar 1 eindevenement

Voordelen & beperkingen BPMN

Voordelen	Beperkingen
Internationaal aanvaarde standaard	Beperkt aantal objecten (geen objecten verbonden met strategie, HR, middelen, wet- en regelgeving)
Link met BPEL	Slechts 1 voorstellingswijze (zwembanen)
Toegankelijk voor de "business"	Geen datamodellen mogelijk
	Geen materiaalstromen (alleen informatie- en sequentiële stromen)