

Wat zijn supply en demand? – Chapter 1

Als er veel vraag is, stijgen de prijzen. Als er veel aanbod is, dalen de prijzen.

Operations management kan op verschillende manieren gebruikt worden:

- Operations management tools kunnen worden toegepast, zodat ze zo efficiënt mogelijk gebruikt worden: het meeste bereiken met de middelen die ze hebben.
- Operations management tools kunnen worden gebruikt om wenselijke afwegingen te maken tussen concurrerende doelstellingen.
- Operations management tools kunnen gebruikt worden om de operations te herontwerpen of herstructureren, zodat de performance over verschillende dimensies verbeterd kunnen worden op hetzelfde moment.

De *current frontier* is de lijn waarop bedrijven zitten die efficiënt werken. De bedrijven die onder de frontier vallen zijn minder efficiënt. Lage kosten en flexibel gaan vaak niet goed samen en dit wordt ook wel een trade-off genoemd.

Hoe zien verschillende processen binnen een organisatie er uit? – Chapter 2

Een *Gantt Chart* geeft de proces stappen weer en hoelang deze duren. Het is een tabel die per tijdseenheid aangeeft wat er gebeurt waardoor er een overzicht ontstaat van welke stappen van het proces hoeveel tijd in beslag nemen.

Proces resources: mensen en gereedschap die nodig zijn voor het proces, bijvoorbeeld machines.

Een proces kan gezien worden als een blackbox, die resources (arbeid en kapitaal) gebruikt om inputs (onbewerkte materialen) om te zetten in outputs (voltooide producten). Dit kan weergegeven worden in een flowdiagram.

Drie fundamentele proces 'meters':

- *Work in proces (WIP)*: Het aantal eenheden in het proces.
- *Flow time*: de tijd die nodig is voor een unit om door het proces te komen.
- *Flow rate/throughput rate*: de ratio waarmee het proces output levert. De maximale ratio die het proces kan genereren is de *capacity* van het proces.

Gemiddelde doorlooptijd = $\frac{\text{Gemiddelde WIP}}{\text{Flow Rate}}$

Little's law: average inventory = average flow rate x average flow time

Flow rate = cost of goods sold

Inventory turns = 1/ flow time

Per-unit inventory costs = annual inventory costs / annual inventory turns

Product process matrix geeft de relatie weer tussen de volume en de variëteit van een product en welke type process bij die combinatie past. Job shop past bij een lage volume en een uniek product. Batch process past bij medium volume en hoge variëteit. Worker-paced line past bij een hoge tot lage variëteit en medium tot hoge volume. Machine-paced line past bij hoge volume en lage variëteit. Continuous process past vooral bij erg hoge volume en een gestandaardiseerd product.

De Groote bespreekt vijf redenen om voorraad te houden:

- *Pipeline inventory*: de voorraad van orders die in bewerking zijn. Dus in het systeem.
- *Seasonal inventory*: buffer voor voorspelbare, hogere afnamen in toekomst. Dus meer ijsverkoop in zomer en daar alvast een voorraad voor maken.
- *Cycle inventory*: er is een voorraad, omdat toelevering of afname batchgewijs plaatsvindt.
- *Decoupling inventory*: er ontstaat een voorraad, omdat processen soms in stappen verlopen. Zo kan er bijvoorbeeld pauze genomen worden.
- *Safety inventory*: buffer voor onzekerheid in de toekomst. Inventory achter de hand houden voor onzekerheden.

Hoe kun je proces capaciteiten evalueren? – Chapter 3

Process capacity: maximum dat een proces kan produceren in een bepaalde tijd.

Process flow diagram: grafische manier om een proces weer te geven.

Boxen: geven een activiteit weer, geeft waarde unit en de activiteit heeft een maximum capaciteit.

Pijlen: geven de route voor de flow aan.

Driehoeken: voorraad, kunnen verschillende soorten en hoeveelheden units in zitten.

Bottleneck: resource met de kleinste capaciteit. Te berekenen door de flow rate per resource te bepalen.

Demand constrained: Als de vraag lager is dan wat ze kunnen leveren, dan is er genoeg capaciteit. Dan wordt er geproduceerd wordt naar de vraag.

Supply constrained: als er meer vraag is dan er geleverd kan worden. Er wordt dan dus op volle capaciteit geproduceerd.

$$\text{Time to fullfill X units} = \frac{X}{\text{Flow rate}}$$

$$\text{Utilization} = \frac{\text{Flow rate}}{\text{Capacity}}$$

De utilization geeft relatief aan hoeveel er nu geproduceerd wordt en tegenover hoeveel er geproduceerd zou kunnen worden. Dit is in procenten.

Er zijn drie redenen waarom een proces niet op 100% produceert:

- Als de vraag lager is dan de levering, zal het proces niet op 100% produceren, maar gewoon op de vraag.
- Als er te weinig input is.
- Als één of meerdere stappen een gelimiteerde beschikbaarheid heeft en dus niet altijd kan produceren.

$$\text{Implied utilization} = \frac{\text{Demand}}{\text{Capacity}} \quad \text{Kan} > 100\% \text{ zijn, dan wordt vraag niet voldaan.}$$

De implied utilization geeft aan wat er door de resources zou kunnen gaan (de vraag) en wat hij werkelijk levert (capaciteit).

Product mix: verschillende types producten worden geproduceerd in één proces.

Vinden van bottleneck bij product mix: er wordt meer dan één soort product geproduceerd, dus niet één flow unit. De vraag en de flow rate zijn dan niet te schrijven als producten/uur.

Oplissing:

- Vertaal de vraag per activiteit in minuten werk per uur.
- Vertaal de cap. per activiteit in minuten per uur
 - $\text{Implied utilization} = \frac{\text{Demand}}{\text{Capacity}}$
- Dan geldt: bottleneck is proces met laagste implied utilization.

Hoe kunnen arbeidskosten worden geïnventariseerd en verlaagd? – Chapter 4

Processing time: tijd om een activiteit te volbrengen.

$$\text{Capacity} = \frac{\text{Number of resources}}{\text{Processing time}}$$

Cycle time: tijd tussen het maken van twee units. Hoe snel er output gecreëerd wordt.

Worker paced line: elke werknemer kan op zijn eigen snelheid werken, er zit voorraad tussen de activiteiten.

Machine paced process: elke machine werkt met dezelfde rate, zodat alle stappen tegelijk klaar zijn. Rate is gelijk aan de bottleneck.

Line balancing: elimineren van idle time. Door proces opnieuw in te richten.

Idle time: leegloop, underutilization.

Tijd door een leeg worker paced process = som van alle procestijden.

Tijd door een leeg machine paced process = aantal resources x procestijd van bottleneck

Time to finish X units starting with an empty system =

$$\frac{\text{time through an empty process} + X - 1 \text{ unit}}{\text{Flow rate}}$$

Performance meting voor menselijke arbeid:

Labor content = sum of processing times with labor

$$\text{Cost of direct labor} = \frac{\text{Total wages per unit of time}}{\text{Flow rate per unit of time}}$$

$$\text{Cycle time} = \frac{1}{\text{Flow rate}}$$

Idle time for single worker = cycle time – processing time of single worker

Om de effectiviteit van een proces met arbeid te berekenen:

$$\text{Average labor utilization} = \frac{\text{labor content}}{\text{Labor content} + \text{sum idle times across workers}}$$

Of

$$\text{Average labor utilization} = 1/\text{aantal werknemers} \times (\text{utilization}_1 + \text{utilization}_n)$$

Om een productielijn meer te laten produceren zijn er drie opties:

- Dezelfde lay-out kopiëren.
- Selectief werknemers toevoegen bij proces stappen.
- Een processtap opsplitsen en door meer mensen laten doen.

$$\text{Requested capacity} = \frac{\text{Number of workers}}{\text{Activity time}}$$

Hoe kunnen projecten worden ingericht en geleid? – Chapter 5

Bij *projectmanagement* richten we ons op 1 product/dienst die in principe eenmalig voortgebracht/geleverd wordt.

Precedence relationships: als er stappen afhankelijk van elkaar zijn. Dus stap A moet voldaan zijn als stap B gedaan moet worden.

Activity-on-node (AON) representation: dit is een weergave die lijkt op de flow diagram, maar dan handiger voor projecten.

Maak een knooppunt (node) door een rechthoek te maken voor activiteit, waarin de naam komt en de verwachte procestijd.

Na een knooppunt komt een lijn die verwijst naar de vervolgactiviteit.

Activiteiten met meer voorganger krijgen verschillende lijnen vanuit deze voorganger naar de activiteit.

Het pad met de langste duur is de *critical path*. De duur hiervan bepaalt de totale duur van het project.

Het bepalen van de duur van een project kan door de *earliest start time (EST)* te bepalen. Dit kan bepaald worden door te kijken naar de eerste tijd waarbij alle voorgangers gedaan zijn. En door de *earliest completion time (ECT)* te bepalen. Dit is de vroegste start tijd plus de duur van die activiteit.

Stappenplan hiervoor:

- Begin met de eerste activiteit. Deze heeft geen voorgangers en begint dus met een EST van 0. De ECT is de duur van deze activiteit.
- Identificeer alle activiteiten die gestart kunnen worden op dit moment (voorgaande activiteit is compleet). De EST voor een activiteit i bereken je door:
 - $EST\ van\ A_i = \text{Max} (ECT\ van\ A_j)$.
 - Waarbij A_j alle activiteiten zijn die input leveren aan A_i .
- Bepaal de ECT van A_i door:
 - $ECT\ A_i = EST\ A_i + \text{Duur van } A_i$

Activiteit i is nu volbracht, identificeer de volgende activiteiten die nu gestart kunnen worden en ga naar stap 2.

Slack time: de totale tijd die een activiteit vertraagd kan zijn zonder de voltooiing van het proces langer te laten duren.

Dit kan door de latest start time (LST) en de latest completion time (LCT) voor elke activiteit te berekenen. Dit doe je te beginnen met de laatste activiteit en zo terug te werken.

Als je het project zo snel mogelijk wil voltooien:

- De LCT van de laatste activiteit = ECT van laatste activiteit.
- Als het project met een bepaalde target tijd voltooid moet worden:
- $LST\ van\ laatste\ activiteit = LCT\ laatste - \text{Duur van laatste}$
- Deze stap steeds herhalen.

Published on *WorldSupporter* (www.worldsupporter.org)

Slack time = latest start time – earliest start time

Twee belangrijkste voordelen van het weten van de slack time zijn:

1. Het later kunnen starten van een activiteit. Dit is vaak handig, omdat activiteiten vaak geassocieerd zijn met kosten.
2. Het is handig om de aanwezigheid van resources te kunnen bepalen. Als deze er soms niet op tijd zijn, kan je het schema aanpassen met de slack time, waardoor je voor het gehele project geen tijd verliest.

Er is vaak onzekerheid bij het bepalen van de duur van projecten. Dit komt doordat:

- De activiteitsduren vaak niet precies te bepalen zijn. Ze kunnen wel langzamer, maar vaak niet sneller.
- Bij meer voorgangers van een activiteit bepaalt de laatst gereede activiteit de EST van de vervolgactiviteit. Bij onzekere activiteitsduren is de kans dan groot dat op zijn minst 1 van die activiteiten lang duurt.
- De onzekerheid kan zorgen dat het kritieke pad verandert.

Rework loops: deels overdoen van een stap, dit beïnvloedt alles erna .

Tussentijds stoppen: go/no-go beslissingen.

Onzekerheid over inhoud: als de vervolgactiviteit niet zeker is.

Een project manager kan de volgende acties ondernemen om de doorlooptijd van het project te versnellen zonder de kwaliteit van het werk of het budget van het project op te offeren:

- Start het project vroegtijdig.
- Manage de omvang van het project.

Crash activities: Vaak wordt het budget verhoogd, om het project sneller te laten verlopen. Maar de vaste kosten van een project zijn vaak hoog, dus wanneer een project erg lang duurt, zal het budget ook verhoogd moeten worden.

Laat critical path activiteiten met elkaar overlappen.

Welke onderbrekingen van de process flow kunnen er ontstaan en hoe hier mee om te gaan? – Chapter 7

Production cycle: het proces dat steeds weer herhaald wordt.

Batch: een verzameling van flow units.

Batching is een noodzakelijk of wenselijk gevolg als er lange setup(omstel)-tijden zijn of hoge omstelkosten. Batching leidt ook tot voorraadvorming voorafgaand aan een processtap.

Als de batch size toeneemt wordt de setup time verdeeld over meer flow units. Daarmee neemt de netto procescapaciteit toe.

De capaciteit van een resource:

$$\text{Capacity given batch size} = \frac{\text{batch size}}{\text{Setup time} + \text{batch size} \times \text{processing time}}$$

De *batch size* is het aantal flow units die gemaakt worden in één cycle voordat het proces zich weer herhaald.

De *setup time* is de totale omsteltijd bij de productie van een batch.

De *processing time* is de totale productietijd die nodig is om een complete flow unit te produceren.

Het is lastig om de interactie tussen batching en voorraad te bepalen. Als een bedrijf twee verschillende soorten producten produceert op één 'loopband', is het handig deze in batches te produceren. Hierbij is dan ook voorraad nodig, om de andere producten op dat moment op te slaan. Wachten is vaak de dominantste activiteit. Hoe groter de batch size, hoe langer de flow units op elkaar moeten wachten voor ze het systeem in kunnen. Dit betekent ook dat er een grotere voorraad capaciteit nodig is.

Mixed-model/heijunka production: werken in batches van één. Dit zou ideaal zijn om te voldoen aan de vraag.

Kiezen van een batch size:

Als de setup tijd de bottleneck is (bij een capacity-constrained process), is het handig de batch size te vergroten, zo wordt de capaciteit groter en daarmee komt er een hogere flow rate.

Als de setup tijd niet de bottleneck is (bij een demand-constrained process), is het handig om de batch size te verkleinen, zodat de voorraad en de flow time afnemen.

$$\text{Recommended batch size} = \frac{\text{flow rate} \times \text{setup time}}{1 - \text{flow rate} \times \text{processing time}}$$

Stappenplan om batch te bepalen:

1. Bepaal de flow rate
2. Bepaal hoe de production cycle eruit ziet met setup en de processen
3. Bepaal hoe lang de setup tijd van een resources is.
4. Bepaal hoe lang alle processing times zijn voor een unit
5. Capacity = $\frac{\text{batch size}}{\text{Setup time} + \text{batch size} \times \text{processing time}}$
6. Capacity (B) = flow rate voor de batch size B.

Published on *WorldSupporter* (www.worldsupporter.org)

$$\text{Recommended batch size} = \frac{\text{flow rate} \times \text{setup time}}{1 - \text{flow rate} \times \text{processing time}}$$

De setuptijd kan gereduceerd worden door *SMED*: Single Minute Exchange of Dies. Deze methode probeert een realistisch doel te maken en zoekt naar potentiële mogelijkheden om de setup tijd te verminderen. Het onderliggende idee van SMED is om alle taken van de setup tijd te analyseren en deze te verdelen in twee groepen, de internal en de external tasks.

Internal tasks: taken die alleen gedaan kunnen worden als de machine stil staat.

External tasks: taken die gedaan kunnen worden terwijl de machine opereert. Dit betekent dat ze kunnen worden gedaan voordat de daadwerkelijke omschakeling plaatsvindt.

Hierbij is het doel om de tijd die nodig is voor de internal tasks te verminderen. Dit kan door bijvoorbeeld te zorgen dat alle materialen klaarstaan wanneer ze nodig zijn.

Economic order quantity: zorgt dat de som van bestel- en voorraadkosten minimaal wordt.

$$\text{Average inventory} = \text{order quantity} / 2$$

$$\text{Inventory costs (per unit of time)} = \frac{1}{2} Q \times h$$

Q = Order quantity

h = holding cost

$$\text{Setup costs (per unit of time)} = \frac{K \times R}{Q}$$

K = setup cost

R = time of ordering cycle

$$EOQ = \sqrt{\frac{2 \times K \times R}{h}}$$

Als C(Q) de som is van de gemiddelde delivery costs per unit time en de gemiddelde holding costs per unit of time geldt

$$\text{Per unit of time costs } C(Q) = \frac{K \times R}{Q} + \frac{1}{2} h \times Q = \sqrt{2} \times K \times R \times h$$

Stappenplan om Economic order quantity te vinden:

- Vindt K, R en h
- Bepaal Inventory costs (per unit of time) = $\frac{1}{2} Q \times h$
- Setup costs (per unit of time) = $\frac{K \times R}{Q}$

$$EOQ = \sqrt{\frac{2 \times K \times R}{h}}$$

De resulterende kosten bepalen door:

$$C(Q) = \sqrt{2} \times K \times R \times h$$

Welke problemen kunnen er ontstaan door wachttijd en hoe hiermee om te gaan? – Chapter 8

Bronnen van variabiliteit:

- Variabiliteit van orders of klanten: aankomsten zijn bijna nooit helemaal regelmatig.
- Variabiliteit in proces tijden: een persoon zal nooit 100 keer een stap op dezelfde manier uitvoeren. Klanten die bediend worden brengen bijna altijd variatie met zich mee, omdat elke klant verschillend is. Bewerkingstijden variëren door de ordermix en door onzekerheid.
- Random beschikbaarheid van resources: bijvoorbeeld machine uitval, ziekte.
- Random routing als er verschillende flow units zijn in het proces: units die allemaal langs andere resources gaan.

Variatie in wachtrij modellen worden weer gegeven voor de coefficient of variation (CV). We beschouwen de CV_a en CV_p (dus 2 bronnen van variatie).

De a slaat op de inter-arrival times (tussenaankomsttijden), de (variabele) tijd tussen twee klant- of orderaankomsten.

De p slaat op de processing times (bedienings/bewerkingstijd), de (variabele) tijd die nodig is om de klant of order te verwerken:

Coefficient of variation = $\frac{\text{Standard deviation}}{\text{Mean}}$

Er zijn altijd wachtrijen, ook als de utilization minder is dan 100%.

Wachtrijtheorie en notatie:

- *Stationary arrival process (generiek)*: klanten arriveren met een bepaald tijdsinterval.
- *Poisson arrival process (exponentieel)*: klanten arriveren met een exponentieel tijdsinterval.

Bij een normale verdeling bepaal je het gemiddelde en de standaard deviatie.

Bij een exponentiële verdeling is gemiddelde = standaard deviatie = CV = 1

G/G/M systeem: Dit is het soort wachtrij en gaat over de aankomsten en processing times.

De eerste letter zijn tussen aankomsttijden. G staat voor generiek (stationary). M voor exponentieel (poisson).

De tweede letter staat voor de 'bedieningsuren'.

De derde letter staat voor aantal bedieners. 1 betekent één bediener, m betekent meerdere bedieners.

G/G/1: generieke aankomsten, 1 bediener.

Utilization = $\frac{\text{Flow rate} = 1/a}{\text{Capacity} = 1/p} = p/a < 100\%$

p = processing time
a = interarrival time

Published on *WorldSupporter* (www.worldsupporter.org)

$$\text{Flow time } T = T_q + p$$

flow time T: tijd in het systeem

T_q = wachttijd in rij

$$I = I_q + I_p$$

I = overall inventory

I_q = inventory in queue

I_p = inventory in process

$$\text{Time in queue } T_q = \text{processing time} \times \left(\frac{\text{utilization}}{1 - \text{utilization}} \right) \times \left(\frac{CV_a^2 + CV_p^2}{2} \right)$$

Resulting inventory:

$$I = R \times T = \frac{1}{a} \times (T_q + p)$$

R = demand rate

$\frac{1}{a}$ = flow rate

G/G/m₊: generieke aankomsten, meerdere bedieners.

$$\text{Utilization} = \frac{\text{flow rate}}{\text{capacity}} = \frac{1 / \text{interarrival time}}{\text{number of resources} / \text{processing time}} = \frac{1/a}{m/p} = \frac{p}{a \times m}$$

a = interarrival time

m = number of resources

p = processing time

$$t_q = \left(\frac{\text{processing time}}{m} \right) \times \left(\frac{\text{utilization}^{\sqrt{2}(n-1)-1}}{1 - \text{utilization}} \right) \times \frac{CV_a^2 - CV_p^2}{2}$$

M/M/1: exponentiele aankomsten, 1 bediener.

$$T_q = p \times \left(\frac{u}{1 - u} \right)$$

$$CV = 1$$

Stappenplan om wachttijden te berekenen:

Bepaal welk systeem

Verzamel de volgende data: aantal resources (m), processing time (p), interarrival time (a), coefficient of variation voor interval (CV_a) en voor processing time (CV_p).

Bereken utilization Bereken utilization $u = \frac{p}{m \times a}$

Bereken T_q

Gebaseerd op T_q kunnen de volgende performance berekeningen gemaakt worden:

- Flow time $T = T_q + p$
- Inventory in service $I_p = m \times u$
- Inventory in queue $I_q = T_q / a$

Published on *WorldSupporter* (www.worldsupporter.org)

- Inventory in the system $I = I_p + I_q$

Pooling betekent dat je m wachtrijen met m afzonderlijke bedieners combineert tot één. Dus ook m keer zoveel aankomsten. Ook in dat geval daalt toch de gemiddelde wachttijd met ongeveer een factor $> m$:

$$t_q = \left(\frac{\text{processing time}}{m} \right) \times \left(\frac{\text{utilization}^{\sqrt{2}(n-1)-1}}{1 - \text{utilization}} \right) \times \frac{CV_a^2 - CV_p^2}{2}$$

Pooling is handig om de wachttijd voor klanten te verlagen zonder extra werknemers aan te hoeven schaffen of het aantal werknemers verminderen en toch hetzelfde aantal klanten te blijven verwerken.

Beperkingen van pooling:

- Minder effect als klant zelf ook al kortste wachtrij herkent en hiervoor kiest.
- Je hebt bedieners nodig die meer soorten bedieningen kunnen doen.
- Klant wil over het algemeen graag een vast aanspreekpunt.
- Er kunnen setups ontstaan bij switchen tussen klanttypen.
- Variatie in bedieningsduren kan toenemen als bedieningen met lange en korte duren gecombineerd worden.

In wachtrijen wordt er vaak geprioriteerd. Dit gebeurt door een kleine stap aan het begin van het proces wanneer klanten arriveren: *de triage step*. Hier wordt informatie verzameld over bepaalde karakteristieken van de arriverende klant. Dit kan op twee manieren:

Processing time dependent priority rules: hierbij wordt gekeken naar de klant met de kortste procestijd en deze krijgt prioriteit. Denk aan de spoedeisende hulp.

Processing time independent priority rules: hier wordt vaak de first come, first served regel gebruikt.

Wachttijden verkorten of perceptie van wachttijden verkorten door:

- Zorg voor spiegels, lectuur, entertainment, invulformulieren
- Splits het proces: snelle eerste stap, volgende rij uit het zicht
- Laat merken dat de klant is gesignaleerd
- Leg oorzaak wachttijd uit, maak bijvoorbeeld prijsvoordeel duidelijk
- First come first service, splits trage en snelle diensten, alleen werkend in beeld
- Breng de (eventuele overschatte) wachttijd in beeld
- Voldoende ruimte om te wachten, koffie serveren
- Zorg dat medewerkers in beeld zijn
- Kies een geschikte wachtrijconfiguratie

De impact van veranderingen op processen “Throughput losses” – Chapter 9

Throughput loss kan gebeuren in twee situaties:

- Er is een gelimiteerde buffer grootte. Als vraag arriveert terwijl de buffer vol is, is dit verlies.
- Flow units zijn onwillig of hebben geen mogelijkheid om lang te wachten voor service, waardoor ze weggaan voordat ze service hebben gehad.

Balking: weg lopen van klanten uit een rij, omdat ze de rij te lang vinden.

Capacity = $\frac{\text{number of resources}}{\text{Processing time}}$

Er bestaan 3 vormen van buffering:

1. Via (*wacht*)*tijd*: de klant wacht op het product/de service.
2. Via (*over*)*capaciteit*: de capaciteit wacht op orders.
3. Via *voorraad aanleggen*: het product wacht op behoefte.

Als een bezette bediener/capacity resource altijd leidt tot een gemiste klant en er dus geen wachttijd is, dan kunnen we de kans dat alle m bedieners/resources bezet zijn bepalen via de *Erlang Loss formule*:

Probability {all m servers are busy} = $P_m(r)$. Opzoeken in appendix B

r = ratio van interarrival time en processing time.

Stappenplan voor Erlang loss formula:

1. Bepaald m , p en a
2. Gebruik appendix B om $P_m(r)$ te bepalen.
3. Bepaal flow rate door:
4. Flow rate = demand rate $\times P_m(r)$
5. $R = 1/a \times (1 - P_m(r))$

Bepaal verloren klanten door:

- Customers lost = demand rate $\times P_m(r) = 1/a \times P_m(r)$

Lean richt zich op de reductie van variabiliteit om tegen minimale bufferkosten te kunnen produceren. Denk bijvoorbeeld aan SMED.

Blocked: als het niet mogelijk is de flow unit door te geven nadat deze afgemaakt is, omdat er geen buffer ruimte meer aanwezig is bij de volgende resource.

Starved: als het stilstaat, omdat de buffer voor de resource leeg is.

Buffers zijn handig om de flow rate van een proces te bepalen. Niet alle activiteiten duren even lang, door buffers neer te zetten, zullen alle resources door kunnen werken, zonder idle time.

Om de omvang van de buffers te beperken (in het belang van capaciteitsbuffers en voorraadbuffers) kan gebruik gemaakt worden van *flexibele buffers*. Dat betekent dat de buffercapaciteit of –voorraad voor verschillende producten kan worden gebruikt.

Published on *WorldSupporter* (www.worldsupporter.org)

Expected sales = $Q - \text{expected lost sales}$

Expected leftover inventory = $Q - \text{expected sales}$ of $Q - \text{expected lost sales}$

Expected profit = $C_u \times \text{expected sales}$

$C_o \times \text{expected leftover}$

Fill rate = $\frac{\text{expected sales}}{\mu}$

De fill rate geeft aan hoeveel er verkocht is ten opzichte van het gemiddelde.

Als we de optimale z-waarde kennen, kunnen we de Expected Lost Sales bepalen:

Expected Lost Sales = $\sigma D \cdot L(z)$

De Normal Loss function (L) is in tabellen te vinden

Bij discrete vraagverdeling en bijvoorbeeld $Q = 5$, geldt

Expected Lost Sales = $\text{Pr}(D=6) \times 1 + \text{Pr}(D=7) \times 2 + \dots$

Want er gaat 1 verloren als de vraag 6 blijkt, 2 bij vraag 7,..

Wat is het order-up-to inventory model? – Chapter 14

Order-up-to model: model wat gebruikt wordt om inventory te managen. Het model bepaalt wanneer er opnieuw besteld moet worden om genoeg voorraad te behouden.

Lead time: tijd die het duurt tot de order ontvangen is. Aangegeven met de letter l . De lead time wordt in periodes gemeten (1 dag, 1 week, 1 maand enz.).

On-order inventory: aantal units die eerder besteld zijn en nog niet gearriveerd. Deze kan nul zijn, maar niet negatief.

On-hand inventory: aantal units die direct aanwezig zijn.

Back-order: totale aantal vraag naar units die er is, maar die nog niet voldaan is. Dit wordt dus alsnog besteld. Ze gaan ervanuit dat er nooit lost sales zijn, omdat er dus nabesteld wordt.

Inventory level = on-hand inventory – back order

Inventory position = on-order inventory + on-hand inventory – back level

Order-up-to level: maximum inventory position die je wilt hebben. deze wordt aangegeven met de letter S . Als aan het begin van de periode gezien wordt dat de inventory position minder is dan de order-up-to level, dan wordt er bijbesteld tot deze gelijk zijn:

Each period's order quantity = S – inventory position

Om de vraag van een langere periode te converteren naar een kortere voor een normale verdeling:

$$\text{Expected demand in short periode} = \frac{\text{expected demand in long period}}{n}$$

Standard deviation of demand in short period = standard deviation of demand in long period / \sqrt{n}

Voor een poisson verdeelde distribution geldt

Standard deviation of a poisson distribution = $\sqrt{\text{mean of the distribution}}$

Stockout: als er vraag is naar een unit, maar er is geen inventory om hieraan te voldoen. Dit is niet hetzelfde als out of stock. Hierbij is er geen inventory meer, maar ook geen vraag. Bij stockout moet er dus ook vraag zijn.

In stock: als aan alle vraag voldaan kan worden met de inventory. *In-stock probability*: de kans dat er in-stock is in een periode. *Stockout probability*: de kans dat stockout plaatsvindt.

Inventory level at the end of the period = S – demand over $l + 1$ periods

$$\text{Stockout probability} = \text{Prob}(\text{demand over } l+1 \text{ periods} > s) = 1 - \text{prob}(\text{demand over } l+1 \text{ periods} \leq S)$$

In stock probability = 1- stockout probability

Expected back order: het verwachte aantal back orders aan het einde van een periode.

Berekenen door:

$$z = \frac{S - \mu}{\sigma} \quad \text{en} \quad \text{expected back order} = \sigma \times L(z)$$

Expected on hand inventory: verwacht aantal units in de inventory aan het einde van een periode.

Published on *WorldSupporter* (www.worldsupporter.org)

$$\begin{aligned}\text{Berekenen door: Expected leftover inventory} &= Q - \text{expected sales} \\ &= Q - \mu + \text{expected lost sales}\end{aligned}$$

Pipeline inventory/expected on-order inventory: gemiddelde aantal inventory on order op elk moment.

$$\text{Inventory} = \text{flow rate} \times \text{flow time}$$

$$\text{Expected on-order inventory} = \text{expected demand in one period} \times \text{lead time.}$$

Omzet management door capaciteitscontroles – Chapter 16

Revenue management/yield management: de wetenschap van het maximaliseren van de inkomsten bij een vast aanbod.

$$\text{Profit} = R \times M - F = \text{Net profit \%} \times R$$

R = revenu

M = gross margin as a percentage of revenue

F = fixed costs

Net profit % = Net profit as a percentage of revenue

$$\% \text{ change in profit} = \frac{((100\% + RI) \times R \times M - F) - (R \times M - F)}{R \times M - F} = \frac{RI \times R \times M}{R \times M - F} = \frac{RI \times M}{(\text{Net profit \%})}$$

Where RI = Revenue Increase

Protection level: aantal dat voor dat tarief of hoger verkocht wordt.

Low fare (tarief) (rl): laagste prijs waarvoor iets verkocht kan worden.

High fare (rh): hoogste prijs waarvoor iets verkocht kan worden.

Booking limit: maximale aantal die voor dat tarief worden verkocht.

Protection level voor de low fare is vaak nul, want als ze hem niet verkopen maken ze nog minder winst.

Booking en protection limits kunnen gebruikt worden bij verschillende soorten bedrijven. Denk bijvoorbeeld aan een hotel dat van te voren kamers verhuurd aan klanten. Met booking limits kun je bepalen wat voor prijzen de kamers moeten hebben, om toch nog winst te maken.

High fare protection level = Capacity – low-fare booking limit

Expected high-fare booking = expected high-fare demand – expected lost sales

Expected number of empty rooms/seats = Q – expected high-fare bookings

We kijken naar de vraag naar services tegen een "high-fare" tarief en daarvoor een "voorraad" reserveren (= protection level)

In plaats daarvan kun we ook kijken naar het aantal *no shows* en daarvoor een aantal plaatsen "reserveren".

De betekenis van de variabelen verandert. Maar de berekeningen blijven hetzelfde:

$$\alpha = \frac{C_u}{C_o + C_u} = \frac{r_h - r_l}{r_h}$$

a wordt gegeven.

$$Q = \mu - z\alpha \times \sigma$$

Voor voorraadberekening:

- C_u (= Cost of underestimating *demand*.)
- C_o (= Cost of overestimating *demand*)

Kansverdeling van de *demand*

Published on *WorldSupporter* (www.worldsupporter.org)

Om de winst te maximaliseren: Bepaal de aan te leggen voorraad

Voor protection level berekening:

- C_u (= Cost of underestimating *high-fare demand*.)
- C_o (= Cost of overestimating *demand*)

Kansverdeling van de *high-fare demand*

Om de winst te maximaliseren: Bepaal protection level voor high-fare klanten

$$\alpha = \frac{C_u}{C_o + C_u}$$

Voor overbooking berekening:

- C_u (= Cost of underestimating *no-shows*)
- C_o (= Cost of overestimating *no-shows*)

Kansverdeling van de *no-shows*

Om de winst te maximaliseren: Bepaal het aantal toegestane overboekingen

$$\alpha = \frac{C_u}{C_o + C_u}$$

Voorbeeld protection level berekenen.

High-Fare demand voor een vliegtuig. Normaal verdeeld, gemiddeld 60, stand.deviatie 15. High fare H=\$69; Low fare: L=\$49

Cost of underestimating High-fare demand; $C_u = H - L = \$20$

Cost of overestimating Demand; $C_o = L = \$49$

$C_u / (C_u + C_o) = 20 / (20 + 49) = 0.29$;

Opzoeken $z_{0.29} = -0.55$ □ $Q = 60 - 0.55 \times 15 = 52$

Kenmerken die bevorderlijk zijn voor bedrijven die booking limits gebruiken:

- Dezelfde unit van capaciteit kan verkocht worden aan verschillende klant segmenten.
- Er zijn verschillende klant segmenten en deze hebben allemaal een andere prijsgevoeligheid.
- Capaciteit kan 'bederven'. Stoel in vliegtuig waar geen ticket voor verkocht wordt, zal geen geld meer opbrengen. Beter verkopen voor een lagere prijs.
- Capaciteit is beperkend. Een stoel in een vliegtuig voor een lage prijs verkocht, kan niet meer verkocht worden aan iemand die een hogere prijs biedt.
- Capaciteit wordt vooruit verkocht.
- Een bedrijf wil opbrengsten maximaliseren, heeft de flexibiliteit om verschillende prijzen te vragen en kan capaciteit van bepaalde segmenten weghouden.
- Een bedrijf heeft concurrentie van een 'discount competitor'.