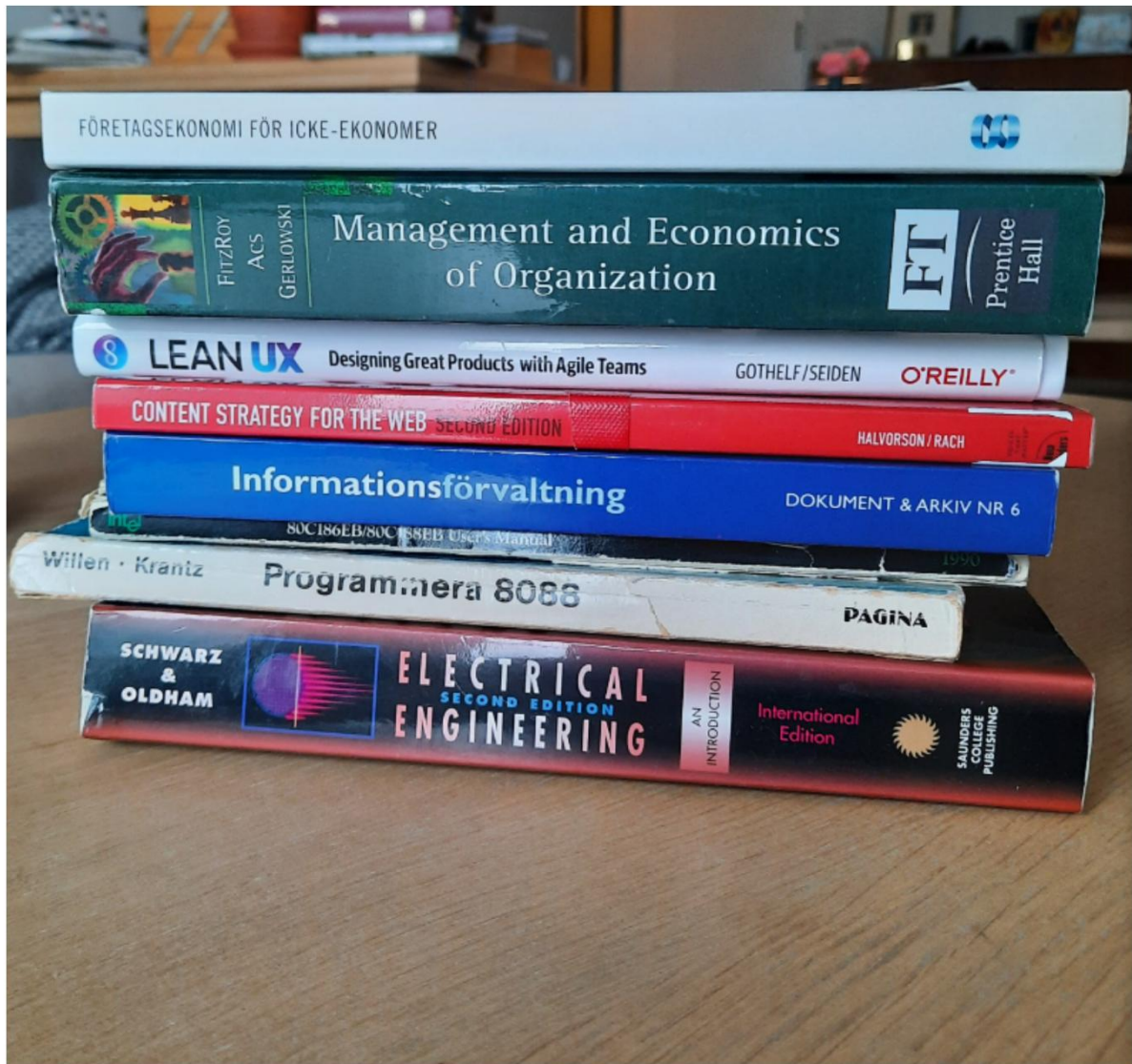


Om hur utveckling av digitala produkter kan illustreras



Detta är förvaltning av en digital produkt. Bilden är konceptuell. Digitala produkter i den här kontexten anses vara offentlig förvaltning, som t.ex ett dokumenthanteringsprodukt, ett intranet, en produkt för politiska ärenden eller produktivitet som M365. I offentlig förvaltning används ofta 'tjänst' isf 'produkt'.

Digitala produkter kan konceptuellt beskrivas som lager eller skiktade system:

Teknik

- Infrastruktur och hårdvara, nätverk, servrar. Lagret, boken, längst ner.
- Operativsystem, databaser, middleware, applikationer. De två böckerna ovanför.

Applikation och Information

- Domänspecifik kunskap och algoritmer och hur användare interagerar, den blå och röda boken. Brukar också kallas backends.
- Frontends i form av användargränssnitt av olika slag, web, mobil, desktopgränssnitt. Den vita boken.

- Dessa två lager, 'Backend' och 'Frontend' är själva applikationen, distribuerad både logiskt, i funktionalitet, informationsstruktur, affärsregler och algoritmer samt fysiskt distribuerad i och med att en backend kan skalas upp och existera i många instanser och en frontend kan distribueras och finnas hos varje användare.

Verksamhet

- En kontext av hur verksamheten är organiserad. Att tillhandahålla produkten som tjänst, ofta där interna kunder köper den och i sin tur tillhandahåller för sina kunder. Olika användarroller som Kund(boende, besökare, företag, samarbetspartners), Medarbetare(Kontorsarbetande, Varierade, Icke-digitala) och särskilda användare som Agent i en kundtjänst eller Handläggare i en socialtjänst. Den organisation som driftar produkten och den organisation som äger produkten och resurser i form av pengar, personal. Mål med verksamheten, lagar och regler. Även att verksamheten upplever att andra digitala produkter hänger samman med denna och att Kunder och Användare har en gemensam upplevelse av användare och ett helt flöde av det de vill, av deras behov, som exempelvis att en produkt för ärenden och bilagor i en dokumenthanteringsprodukt fungerar ihop, som exempelvis 'att hitta information' och att detta beskrivs på ett intranet.

Affären

- Den översta boken representerar ekonomin, att produkten går ihop ekonomiskt. Mycket grovt generaliserat kan sägas att Kostnad: all design av en produkt och Intäkt: alla funktioner i en produkt. Dessa måste balanseras för att produkten skall gå ihop, annars blir det entusiastprojekt/open source. Den översta boken i stapeln.

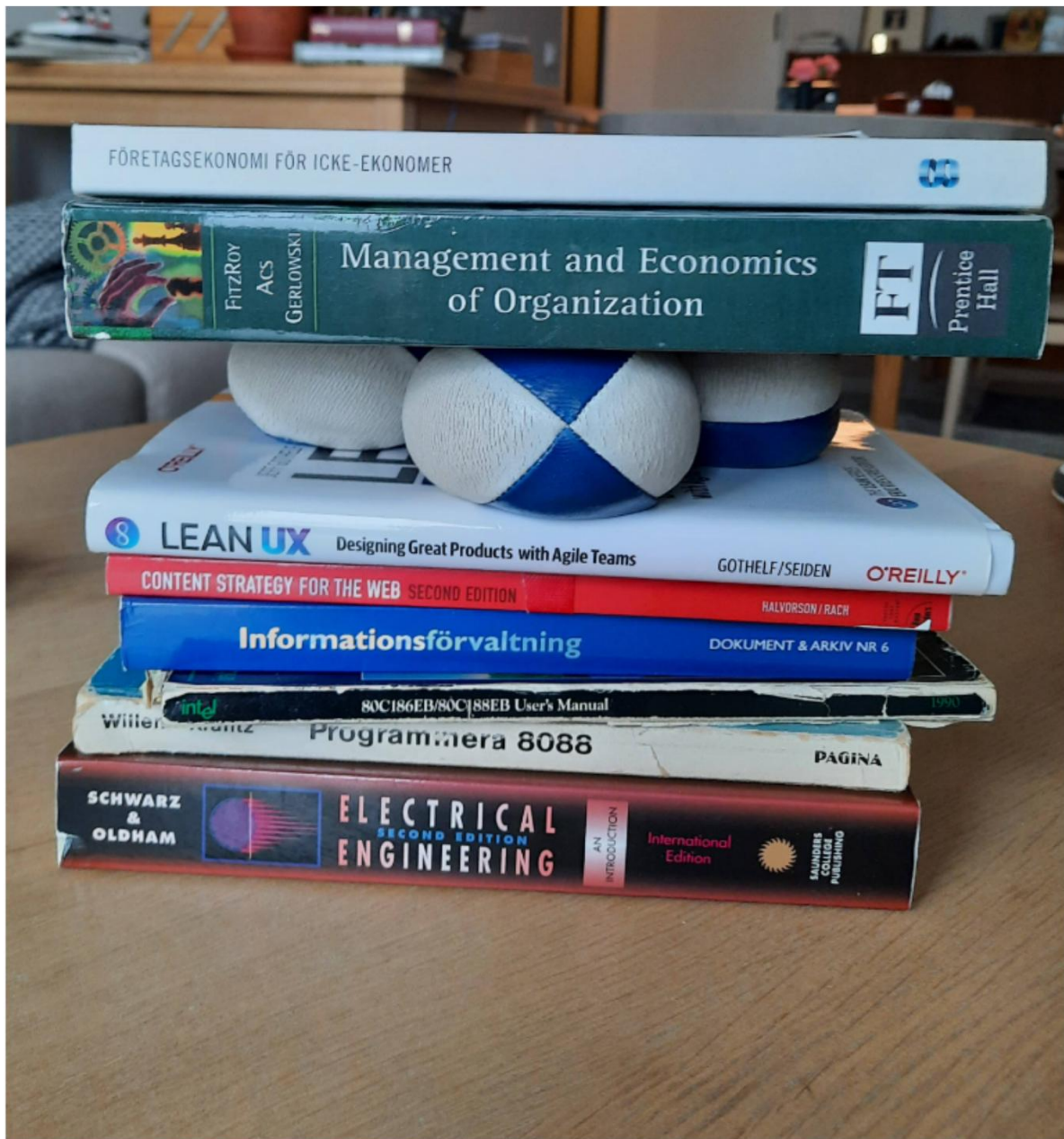
Det finns alltså flera lager och för detta krävs kompetenser som fått produkten att hänga ihop. Detta representeras av att böckerna är någotsånär i en prydlig hög, den är inte helt perfekt, men nära nog.

- Arkitektur och UX/Tjänstedesign ser till att behov från användare(outcome) och verksamhet(impact) ensas med den tekniska lösningen(features). Att det blivit rätt produkt. Detta kan sägas vara statiskt. Det sker ingen förändring.
- Förvaltning ser till att produkten fortfarande är rätt. Dvs att produkten fungerar, är stabil, uppdateringar och säkerhet tillgodoses.

Trouble in paradise



Här har dock något hänt. Det har kommit in nya behov någonstans runt själva applikationen och systemet. Det kan vara en ny uppdatering med funktionalitet som kanske måste införas, eller ett avtal eller licens som förändras och kanske måste ett system bytas ut. Det kan vara säkerhetsmässiga aspekter.



Här har något annat hänt. Det har kommit in behov från användare. De kan vilja ha ny funktionalitet. Eller en teknik, tänk vad som hände med 'Silverlight' är deprecated och gränssnitt måste skrivas om. En ny trend, tänk när 'Responsive design' slog igenom.

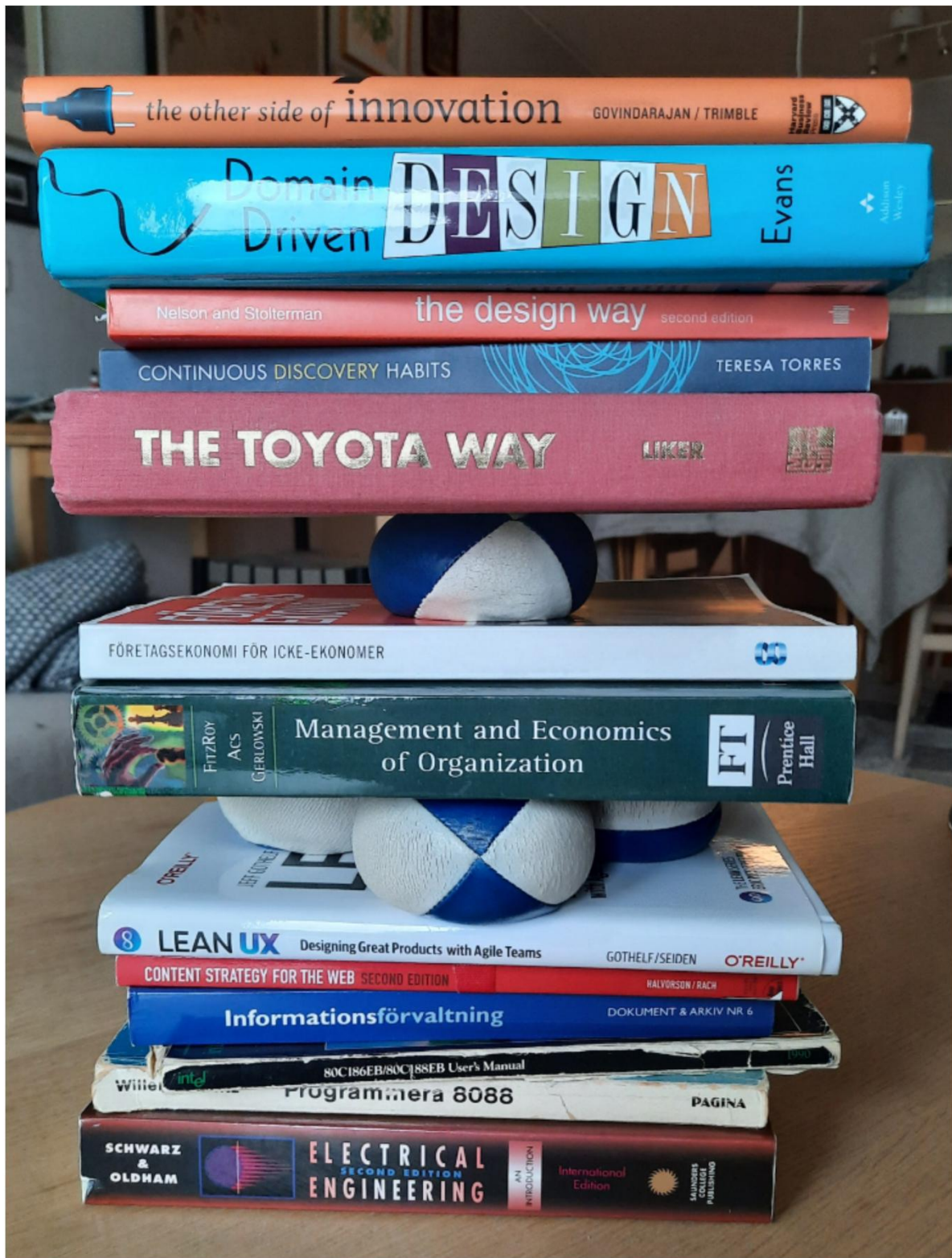
Den svarta boken, tredje från botten, är nu lite på sned. Det kanske går att tweaka till och klara sig utan större förändringar.



Och här har mål eller regler som styr verksamheten förändrats. Tänk GDPR eller nya mål och riktlinjer utifrån politiska förändringar.

Eftersom de olika lagren hänger samman så förstår vi att oavsett var ett behov uppstår så kommer det att påverka alla delar. Hela produkten, dess leverans som tjänst, dess organisation, dess ekonomiska förutsättningar och hur en är ensad mot vad användare, verksamhet vill ha och även övergripande mål samt andra digitala produkter och alla deras aspekter, hänger samman.

Hur Utveckling av digitala produkter ser ut jämfört med Förvaltning



Uppstår ett nytt behov och måste vi ta tag i det så lämnar vi Förvaltningsvärlden och går in i Utvecklingsvärlden.

Utveckling är fundamentalt, by definition, helt olika mot förvaltning. Det är vanligt förekommande att verksamheter försöker hantera utveckling i en organisation, på samma sätt, som förvaltning.

- Utveckling är per definition oförutsägbar medan förvaltning är förutsägbar. Den kan sägas:
- Det som flyter i processen är kunskap, medan i förvaltning rör det sig om levernas, KPI'er. Utveckling är närmare OKR'er.
- Takten i utveckling är i iterationer av lärande, tillväxt i funktionalitet, utforskande ibland måste man gå tillbaka. Förvaltning är inkrement, upprepning, samma sak möjligen med förbättring i form av lägre kostnad, tid osv.
- Utvecklingsprocessen består av designtänkande med experiment och praktisk kunskap med syfte att maximera lärande och där input är hypoteser. Förvaltningsprocessen behöver av relativt fasta steg i tillverkning eller leverans av tjänsten och där input är beställning.
- (Läs mer: irm.se/artiklar/utveckling-ar-larande/)
- I viss mån även att verksamhet och teknik har olika filosofi, talar olika språk. (Läs mer: Almaden, IBM)
- Fundamentala skillnader och ny organisation krävs, ibland helt ny verksamhet/bolag. (Läs mer: 'The other side of innovation')

Vissa saker är gemensamma:

- Kostnaden för designen och intäkterna från funktionalitet är modellmässigt samma men kostnaden kan öka eller minska och utifrån kundkrets eller erbjudande kan intäkterna öka eller minska.
- Domänen är normalt samma. Domänkunskap kan fördjupas men ingen ny domän behöver läras in. Undantag om man ger sig in i en ny domän givetvis.

Vissa saker är nya:

- De lager som tillkommit representerar konceptuellt design och process och ledning.
- Design kring hur behov fångas och lösningar utifrån ett oändligt antal möjligheter utifrån olika constraints
- Utvecklingsprocessen som kan vara traditionell, vattenfall, agile eller en kombination.
- Ledning som sätter ramar kring policies, förutsättningar, principer. Budget i tid och pengar och resurser.

Så hur gör vi då



Denna bild illustrerar hur alla olika delar i 'Utvecklingsvärlden' ser ut, konceptuellt och förenklat.

Ovanpå behoven som uppkommit, på olika lager, högst upp affär och verksamhetsmässig. Där hamnar nu lager av hur utvecklingsprocessen ser ut, hur design av digitala produkter går till, hur en organisation kan sättas upp för att ta sig in i 'Utvecklingsvärlden' (den orange och mörkblå boken högst upp).



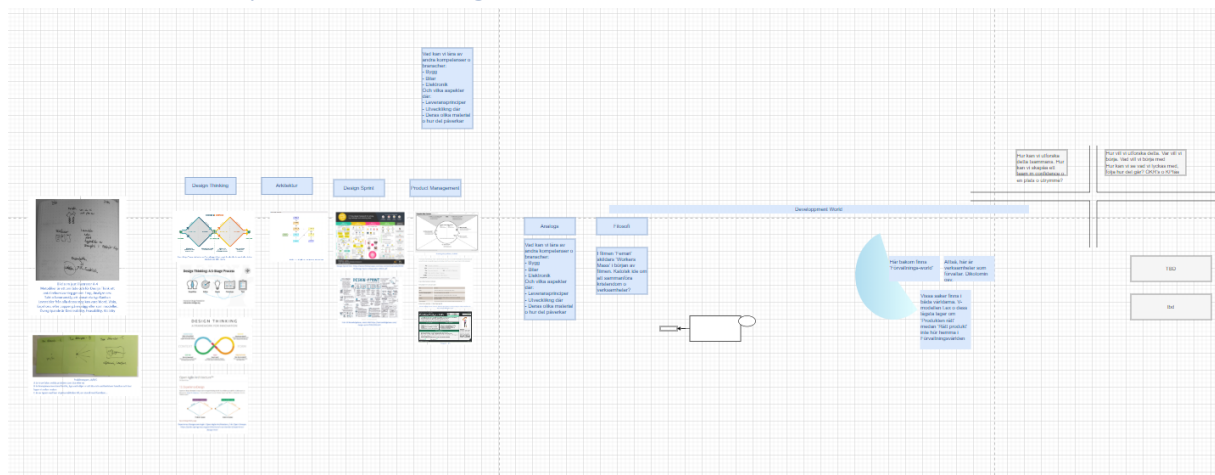
Detta är så många aspekter att det kan anses omöjligt. Det behövs därför en strategi. Se beskrivningen av hur något händer när man antingen börjar få för många behov, jongleringsbollar, eller när man förstår dessa, för många mindre behov och andra krav som påverkar, pennorna. Eller insikten om att förståelse av affären och saker som påverkar den, kan vrida just det lagret, den boken så att flödet genom lagren hamnar snett, rör man ett lager, tappar eller vrider så måste alla andra hänga med, detta handlar om relationer, beroenden, förståelse för helhet och att managera genom takt, täta prov och tidig upptäckt.

- om man återgår till bilden av bokstapeln, så att inte stapeln tappar över av alla behov som kommer in, så behövs ett sätt att få igenom behov till fungerande tekniska lösningar utan att högen av behov, pennorna, bara byggs på och inte kan betas av. Detta skär tvärs igenom stapeln, från att förstå behov, bryta ner, lära sig, prova olika lösningar, besluta och leverera efter att arkitektur ensat både teknik, behov, affär.

- Om man vrider en av böckerna, om antingen t.ex affären börjar upptäcka andra mål, eller det kommer in nya personer, beställare och bestämmare med andra prioriteringar, så måste hela stapeln hänga med, ända ner till det lägsta lagret.
- När man utforskar ett behov, jongleringsbollarna, så kommer det oundvikligen nya behov, nya aspekter, fler småbehov, saker man inte tänkt på, problem längre ner i lagren eller aspekter från sidan som säkerhet. Detta illustreras av pennorna som spawnats ut från jongleringsbollarna.
- Obalans mellan lagren. Detta kallas 'timglaset' eller 'det blågrå virrvarret' och är en variant av 'the design squiggle', men på vertikalen. Se nedan. Den har att göra med att när man bryter ner ett behov, en jongleringsboll, så uppstår inte bara en massa nya mindre behov, problem, pennorna på samma nivå, utan de växer och blir normalt fler ju längre ner i lagren man kommer. Och dessa måste förstås, utifrån det lager där man är. Man kan inte hoppa över ett lager. Det leder till oförståelse, ökade risker för fle uppfattat behov, att man inte förstått hela behovet, att man inte förstått sidoeffekter eller att aspekter från sidan som säkerhet inte adresseras. Ofta förekommande i Förvaltningsvärlden är att man fångar behov och går med dem direkt till IT-avdelningen med resultat att lösningen blir fel och att det blir fel lösning, och samtidigt växer mängden behov ovanifrån tills hela organisationen stannar upp. Detta har också att göra med 'systemics', komplexitet och beroenden som måste hanteras genom att bryta ner, isolera så problemen blir komplicerade istället. Matematiskt kan detta bevisas med 'Littles Law' och via systemics kan man inte enkelt beskriva men i alla fall förstå o-intuitiva förlopp av kaotisk natur, enklast illustrerat med företelsen 'staue' på Tyska motorvägar och som uppkommer när ett system pressas mot sin kapacitetsgräns. DevOps är ett sätt att hantera detta, återigen, detta är inom domänen för Utvecklingsvärlden och där Förvaltningsvärlden inte har redskap att hantera detta beroende på att en utvecklingsprocess per definition är oförutsägbar och tesen att bilden nedan illustrerar hur den faktiskt ser ut 'the Design Squiggle'.



Ett sätt att skapa 'Utvecklingsvärlden'



Detta är en allra första version av en bild som illustrerar 'Utvecklingsvärlden'. Det den visar är vad som ingår, från hur utveckling styrs och leds och utförs, hur team byggs och fås att fungera ihop och leverera och hur prioritering och värde kan guida i hur oförutsägbarhet kan manageras så att en lösning ändå kommer fram inom budgetramar och uppfyller de mål som satts upp. De olika kompetenser som behövs generellt kring digitala produkter, som infrastruktur och operativsystem och databaser och middleware. Utvecklingsramverk, programspråk. Förståelse för domänen, användare, verksamheten. Arkitektur som ensar och stabiliserar designen. Design av hela upplevelsen, tjänste och enterprisedesign som gör att produkten fungerar i sitt sammanhang i både hela verksamheten och i samhället. Överlämnningen av den designade produkten, tillbaka in i Förvaltningsvärlden. Vissa av dessa delar är mera av gränssytor, så är t.ex verksamheten och domänen mera en gränssyta som behöver manageras, medan t.ex projektledning kan vara något som förekommer i många andra sammanhang i en verksamhet medan specifika delar som att programmera i Python enbart förekommer just i digital produktdesign. Dessa gränssytor är den ljusblå tonade sektorn lite till höger i bilden.

Detta är så många aspekter att det kan anses omöjligt. Det behövs därför en strategi. Se beskrivningen av hur något händer när man antingen börjar få för många behov, jongleringsbollar, eller när man förstår dessa, för många mindre behov och andra krav som påverkar, pennorna. Eller insikten om att förståelse av affären och saker som påverkar den, kan vrida just det lagret, den boken så att flödet genom lagren hamnar snett, rör man ett lager, tippar eller vrider så måste alla andra hänga med, detta handlar om relationer, beroenden, förståelse för helhet och att managera genom takt, täta prov och tidig upptäckt.

Strategier är:

- Agila metodiker. Kaizen. Scrum. Kanban.
- Designmetodiker. Design thinking är normalt för linjärt, särskilt 'Design Sprint' eller 'Innovationsguiden' och behöver en mera iterativ approachering. Men har många bra element, som fokusgrupper, principer med diverge/converge och praktiska tekniker för att komma förbi analysis paralysis(crazy eights m många flera).
- Lean Startup. Lärande, validated learning. Angränsar till designtänkande och gör detta mera iterativt och organiserar en verksamhet för lärande.
- 'Mini rogue' där man hela tiden går igenom alla aspekter, alla lager, men lite i taget. Man driver dock hela tiden mot en komplett iteration ända till en praktisk produkt, om än prototyp av något slag.
- 'T-Shape' eller 'Spike'. Detta är en djup vertikal men bara genom ett litet område i varje lager. Principen är att om man går på djupet i en fråga, och lär sig 80% av problemet där så faller det ut 20% lärande i andra lager och detta kan vara tillräckligt och/eller ge riktning för nästa iteration.

Förslag på att hantera 'Utvecklingsvärlden'

Först måste man inse att problemet ser ut något åt det här hållet, det är en konceptuell bild, abstrakt osv. men i stort. Det är lager som hela tiden förskjuts i förhållande till varandra, där behov uppstår emellan och gör att hela stapeln blir instabil men måste hållas samman, när förutsättningar förändras eller behoven växer, jongleringsbollarna blir fler, böckerna i stapeln börjar vridas i förhållande till varandra när förutsättningar förändras och alla delar måste följa med.

Offentlig förvaltning tänker ofta i 'tjänst' isf 'produkt' vilket kan orsaka en del begreppsförvirring gentemot branschen som är vanare att tänka i produkter som bygger tjänster. Den ekonomiska modellen är också ofta olika och skapar förvirring.

Metodiker som SAFE och andra försöker adressera de här problemen, men blir ofta för stort och är lite av en 'take it', när en annan grundtes är att alla verksamheter är olika, de är individer, bestående av kultur, medarbetare, historia osv. och vad som funkar på ett ställe behöver inte alls funka i ett annat.

Ffa måste man inse att det är skillnad på 'Utvecklingsvärlden' och 'Förvaltningsvärlden'.

Ansätser finns och där kan man hämta erfarenhet:

- Arkitektur i hela spannet är bra, men missar delar som designtänkande och har emellanåt inte alltid förståelse för 'materialet' dvs systemutveckling, b.la beroende på att arkitekter inte alltid har teknisk bakgrund eller iallafall inte bred eller djup erfarenhet.
- Edgy23 är ett bra ramverk för enterprise design. Värt att titta närmare på.
- Service Blueprint är kanske det mest praktiska verktyget och mest värt att titta närmare på.
- 'Apophthegmata'-konceptet för att fånga principer, metodiker och göra enkla verktyg som är skalbara, enkla att ta fram och bygga vidare på. Detta är också en framkomlig väg.
- 'Analoga hjälpare', en slags rollspels-automata är också en väg som bör provas.

En ansats kan vara att inte skapa stora ramverk, stora organisationer, göra om allt utan istället gå bottom-up med praktiska tips, lite som 'Arkitekturprinciperna' i ADD'n i Staden. Praktiska verktyg som skapar skills utifrån metodiker och mindset man inte alltid behöver ha full koll på, verktygen går att använda ändå. Man kan då rusta medarbetare som redan finns och behöver inte så många nya resurser. Mentorskap, 'arkitekten som terapeut' och 'användarens och verksamhetens advokat' är också bra begrepp men illustrerar samtidigt att det är verksamheten själva som måste komma fram till hur de skall organisera sig samt vad deras digitala produkter faktiskt skall göra, för dem, i deras typ av domän – en mentor /arkitekt/tjänstedesigner kan bara hjälpa dem att själva komma fram till detta.

Så hur kan vi göra praktiskt

'A conceptual heap of book' eller 'A conceptual heap of plastburkar' illustrates övergripande sammanhang. De utgår också från att försöka hitta sätt som når fram till olika typer av åhörare, intressenter, läsare, i verksamheter som behöver rustas. Denna typ av övergripande beskrivningar behövs, det finns också andra som behövs, men denna illustrerar ändå flera olika, kanske de grundläggande koncepten. Lager, ramverk, vikten av användare o deras behov, dynamiken och utvärdering/förändring över tid, systemics både i komplexitet o att lager o alla andra delar hänger samman o påverkar varandra.

Traditionellt har denna typ av konceptuella sammanhang, som kan skapa en intuitiv förståelse, sedan alltid ha behövts ackompanjeras av praktiska sätt och verktyg för att tillämpa, i verkligheten. Att hantera abstraktion, komplexitet, att tillämpa erfarenhet praktiskt osv. Dessa växer lätt till omfattande ramverk som nästan utan undantag kollapsar av mängden roller, mängden leverabler, uppföljning och allt annat som skymmer det viktiga – leverans av rätt produkt och produkten rätt.

Ett annat sätt är att gå från andra hållet. Den agila rörelsen växte fram som en motvikt till ett av dessa stora ramverk, RUP, i början av 2000-talet, men den agila rörelsen har sedan delvis också kollapsat under tyngden av ramverk som SAFE, eller ottydlighet, oförståelighet hos intressenter i verksamheten som levererats förenklade budskap och utan en grundläggande förståelse för vad utveckling innebär, eller kunskap om materialet, 'digitalt', för att skapa produkter.

En annan brist är att ramverk tenderar att utgå från att de själva kan lösa alla problem, på ett eget sätt, av olika skäl, kommersiella, ideologiska, att man inte kommer överens med andra ofta starka personligheter i branschen.

Ett annat sätt som vi tänker prova är därför att anamma dels färdigheter och ideer och verktyg från alla delar som behövs i utvecklingsarbete, design, sysdev, arkitektur, UX, projektledning, den framväxande 'continuous product management'. Och att

- Lägga till en tradition från den framväxande antika bildningen under 2-300 talet och framåt som byggde på djup kunskap från antik akademi men lade till enkelhet, självständiga/high cohesion/low coupling tänk och praktisk kunskap. Många av dessa egenskaper har sedermera gått förlorat både i akademi, white/blue collar tänk i västvärlden. Men har alltid, genom årtusendena levt kvar i monastisk kristendom och som börjat få en renässans i västlig kristendom mera brett.
- Genom t.ex 'analoga hjälpare', 'tänkespråk'
-

Övningar

Bygg din egen stapel av böcker för Förvaltningsvärlden

1. Hur skulle en stapel av böcker som representerar digitala produkter i din organisation se ut?
2. Var ärlig. Vilka lager finns faktiskt? Är de förstådda? Vad representeras de av och hur ser erfarenhet ut, är det bara något från en konferens eller finns praktisk kunskap kring de olika lagren?
3. Är lagren, böckerna, ensade med varandra. Finns det en förstådd och beskriven arkitektur så att man kan följa en linje mellan lagren, från behov och användare ner till lösning.
4. Hur ser denna stapel ut jämfört med andra. Dela med er så kan vi presentera olika staplar.
5. Lagg märke till hur en dialog spontant uppstår när du behöver gå runt i verksamheten och intervjua och fånga lager och representativa böcker.

Illustrera hur behov uppstår i denna stapel

1. Illustrera behov med något lämpligt objekt. Bollar, godis, klotsar, pennor.
2. Var ärlig och illustrera var behoven uppstår, var de hamnar, i vilket lager.
3. Var finns flest behov, högst upp, från beställare, från användare mera i mitten, eller längst ner runt tekniska lager och applikationer.

Bygg din egen stapel för Utvecklingsvärlden

1. Hur ser en stapel ut, vilka lager finns för att ta hand om de behov som illustrerats?
2. Var ärlig. Vilka är typiska böcker. Varifrån kommer de, vilka förmågor finns i verksamheten, vilka roller, vilka personer.
3. Hur ser denna stapel ut jämfört med andra. Dela med er så kan vi presentera olika staplar.
4. Hur ser ny dynamiken ut när behov börjar omsättas i utvecklingsprocessen, design av lösningar. Är det en jämn fördelning, eller mera normalt, fler och fler behov uppstår när man förstår dem och kompletterar med andra aspekter.
5. Blir det övervikt någonstans, riskerar ett lager att tippa eller hela stapeln att välta av t.ex för många behov högt upp eller i ett mellanlager och som inte hinner realiseras utan blir kvar och växer.
6. Blir det obalans, förskjuts vissa lager, eller vad händer när de vrids, dvs förutsättningar förändras, behov vi inte sett förut, sidoeffekter eller att organisationen inte förmår, inte hinner, kompetenser saknas eller får nya insikter som gör att en bok börjar vridas, följre de anrda i stapeln med?