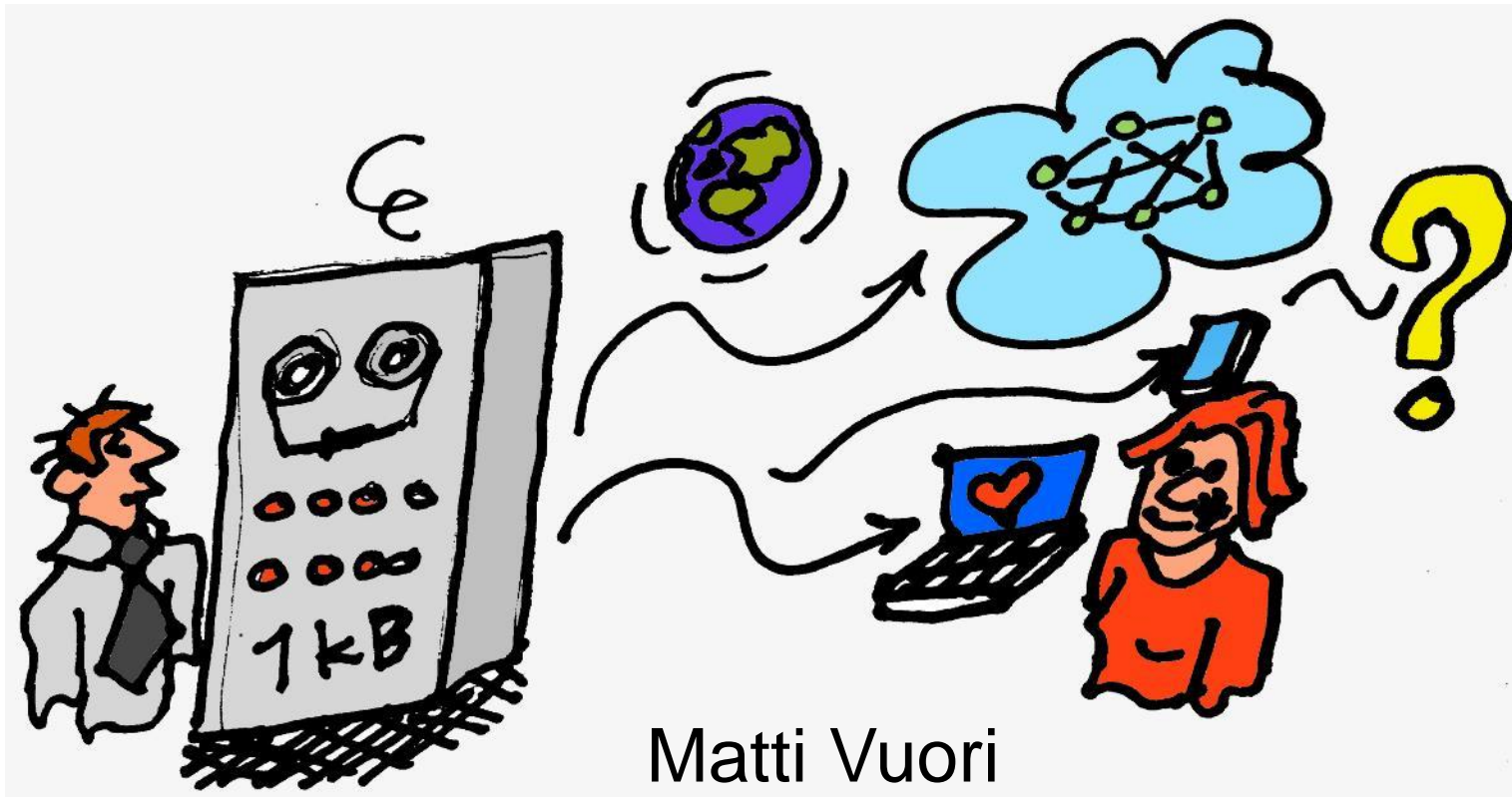
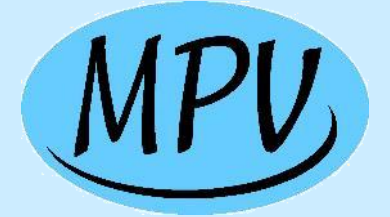


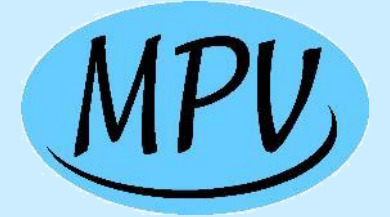
Mistä tulemme? Miten? Minne menemme? Tietojärjestelmien ja niiden kehittämisen historiasta ja muutoksista viisautta tulevaisuuteen





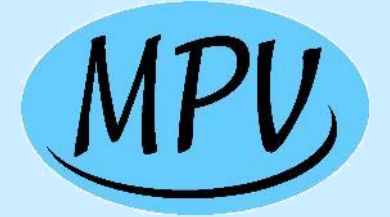
Sisällysluettelo 1/4

<u>Motto</u>	<u>6</u>
<u>Johdanto</u>	<u>7</u>
<u>Historian tarkastelutapoja</u>	<u>8</u>
<u>Tulevaisuuden muotoilu</u>	<u>10</u>
<u>Muutosten synty</u>	<u>11</u>
<u>Isoimmat kiireen tunteet eri aikoina</u>	<u>12</u>
<u>Miten reagoidaan havaittuihin muutoksiin?</u>	<u>13</u>
<u>SYSTEMIEN PIIRTEET</u>	<u>16</u>
<u>Näkemykset kehitettävistä systeemeistä</u>	<u>17</u>
<u>Laatukäsitys rikastuu</u>	<u>19</u>
<u>Kompleksisuus kasvaa</u>	<u>20</u>
<u>ATK kantaa merkityksiä</u>	<u>21</u>
<u>Tietotyöntekijä ja merkityksellisyys</u>	<u>22</u>
<u>Suhde käyttäjiin syvenee</u>	<u>23</u>



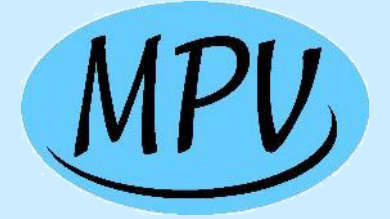
Sisällysluettelo 2/4

<u>Käyttöliittymätyypit</u>	26
<u>Tehdäänkö ihan oma tietojärjestelmä, räätälöity vaiko vain konffataan valmis</u>	27
<u>Vai kaikki palveluna?</u>	28
<u>Avoimuus oletukseksi</u>	29
<u>Suunnittelunormit ja standardit</u>	30
<u>Missä dataa prosessoidaan?</u>	31
<u>Sovellusten määrä ja kokonaisarkkitehtuuri</u>	32
<u>ATK-älyn muodot muuttuvat</u>	33
<u>Datan käyttö</u>	34
<u>Datan arvo oivallettu</u>	36
<u>Kyberturvallisuus</u>	37
<u>Päivitystahti</u>	39
<u>TYÖTAVAT SYSTEEMIEN KEHITTÄMISESSÄ</u>	40
<u>Prosessimallit</u>	41



Sisällysluettelo 3/4

<u>Ohjelmointikielet</u>	44
<u>Arkkitehtuurityö</u>	45
<u>Tapoja perussuunnitteluun</u>	46
<u>Tieto onnistumisesta</u>	48
<u>Rytminen reflektio</u>	49
<u>Testaus ja laadunvarmistus</u>	50
<u>Testausaika</u>	51
<u>Johtaminen</u>	52
<u>Prosessinormit ja prosessi-standardit</u>	53
<u>Yhteistyö organisaatiossa</u>	54
<u>Lähellä vai etänä</u>	55
<u>Automaatiota eri paikoissa</u>	56
<u>SYSTEEMIEN TEKIJÄT</u>	57
<u>Erikoistuminen</u>	58



Sisällysluettelo 4/4

Työryhmät, tiimit	59
Ideaali organisaatio ja eri aikojen huolenaiheet	60
Resurssointi	61
Kehittäjät ja sukupuoliterveys	62
Osaaminen	63
Osaamisen ja ongelmanratkaisun jatkuva tuki	65
PARI LINKKIÄ	66

**Jos muutos on ainoa pysyvä asia, kannattaa tarkastella sitä ensimmäisenä ja tarkimmin.
Näin saadaan eniten tolkkua asioista.**

- Tulevaisuuden ja organisaatioiden tutkijat korostavat historian tuntemisen tärkeyttä.
- Kun tietää mistä ja miten on tultu nykypäivään, saa eväitä tulevaisuudenkin miettimiseen.
- Saadaan perspektiiviä ja osoitusta siitä, että asiat voivat olla toisin ja muutos on aina mahdollinen kaikissa asioissa. Varsinkin murrosten ja kriisien aikana on hyvä miettiä oman toimintaympäristönsä olemusta.
- Eikä tarvitse toistaa vanhoja virheitä.
- Tässä koosteessa käymme läpi ATK-maailman muutoksia, niiden luonnetta ja mitä niistä voisi oppia. Aikaperspektiivi alkaa 1960-luvulta, kun tietokoneet ja digitalisaatio alkoivat puskea maailmaamme. Kontekstina on eniten tietojärjestelmien kehittäminen ja käyttö.

Historian tarkastelutapoja 1/2

- Historiaa katsellaan yleensä kolmella tyylillä:
 - Historiikit.
 - Muistelmat.
 - Analyyttiset yhden tai useamman teeman tarkastelut.
- Tässä toteutetaan kohtalaisen analyttistä tyyliä, mille on ominaista kokonaisuuden jäsenitys elementteihin tai alueisiin ja sen jälkeen niiden tarkastelu.
- Tämä on tärkeää, sillä usein tartutaan vain yhteen teemaan kerrallaan (ketteryys, tekoäly, resilienssi...)

Historian tarkastelutapoja 2/2

- Eriyisen tärkeää tämä työ, jos aiotaan ymmärtää ja kehittää jotain toimialaa yms. (Kuten vaikka Pitkyssä)
- Ilmiöhavaintoja pitää käsitellä datana: suodatus, yleistys (luokittelu), siivoaminen....
- Ideana ei ole olla kattava tai eksakti, vaan esimerkinomaisesti ilmiöitä tiivistävä.

Tulevaisuuden muotoilu

- Miten historiasta päästään tulevaisuuteen? Ensimmäisenä pitää jäsentää tarkasteltavaa kokonaisuutta. Eli mistä se koostuu?
 - Jäsennysapuna voi olla PESTLE (Political, Economic, Societal, Technical, Legal, Ecological etc...) tai kuten tässä dokumentissa kontekstista bottom up –haettu jäsennys.
- Sitten tehdään kokonaisuuden ja elementtien mahdollisten muutosten hahmottelua.
 - Jäsennys auttaa siinä-
- Skenaarioiden luominen: todennäköinen, pahin mahdollinen, paras mahdollinen muutosten yhdistelmä.
- Sitten miettiminen mistä tiedetään missä skenaariossa mennään ja pitäisikö vaihtaa toiseen.

Muutosten synty

- Muutoksia syntyy ajan mittaan monella tavalla.
 - Reaktio ongelmiin
 - Reaktio emergenttiin mahdollisuuteen
 - Jatkumo, orgaaninen kehitys
 - Kahden ilmiön yhdistyminen uudeksi
 - Uudelleenajattelu
 - Pakko
- Osa on sisäsyntyistä, osa juontuu ulkopuolelta, osaa halutaan, osaa taas ei...
- *Mutta kaikki mikä tapahtuu tapahtuu ja kaikenlaista voi tapahtua! Ideana on nykyään olla muutosmyönteinen hyvälle asioille ja tukea rakenteilla ja toiminnalla hyviä muutoksia.*

Isoimmat kiireen tunteet eri aikoina

- Jokaisella aikakaudella on dominoivia isoja asioita, joista tulee "sense of urgency" (muutosjohtamisguru Kottlerin termi).
- Tällaisten kanssa on eletty:
 - Tietokoneiden käyttö -> osaaminen -> ammattilaisten kouluttaminen -> kansalaistaidot -> kyberturvallisuus -> vastuullisuus
- *Tämä muistuttaa siitä, että maailma muuttuu. On nähtävä, mihin ollaan matkalla ja hyväksyttävä haasteet mieluummin etukenossa kuin jälkijunassa.*

Miten reagoidaan havaittuihin muutoksiin? 1/3

- *Kaikki pitää miettiä oman tai kehitettävän kontekstin kannalta, mutta...*
- Jos on nähtävissä pysyvä trendi:
 - Ehkä se kannattaa ottaa vakavasti
- Jos joku asia korostuu ihan liikaa:
 - Koitetaan luoda sille tasapainoa
- Jos on huojuntaa:
 - Pitää miettiä missä mennään
- Jos on uusi ilmiö:
 - Pohdittava sen olemusta, onko se päivänperho vai kantava mahdollisuus tai ilmaantunut tarvealue



Miten reagoidaan havaittuihin muutoksiin? 2/3

- Kun kaksi asiaa muuttuu:
 - Voiko niistä luoda uuden kolmannen?
- Jos jokin asia on kadonnut:
 - Ehkä unohdetaan se itsekkin? Tai tuodaan takaisin
- Jos nähdään virhe tai laiminlyönti:
 - Ei tehdä sitä omassa toiminnassa
- Jos jostain puhutaan paljon:
 - Mietitään mitä on puheen takana, mistä se kumpuaa? Yhdytäänkö puheeseen puhutaanko jostain muusta?
- Jos jostain ei puhuta:
 - Ehkä aletaan puhua juuri siitä?

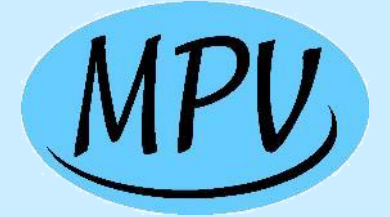


Miten reagoidaan havaittuihin muutoksiin? 3/3

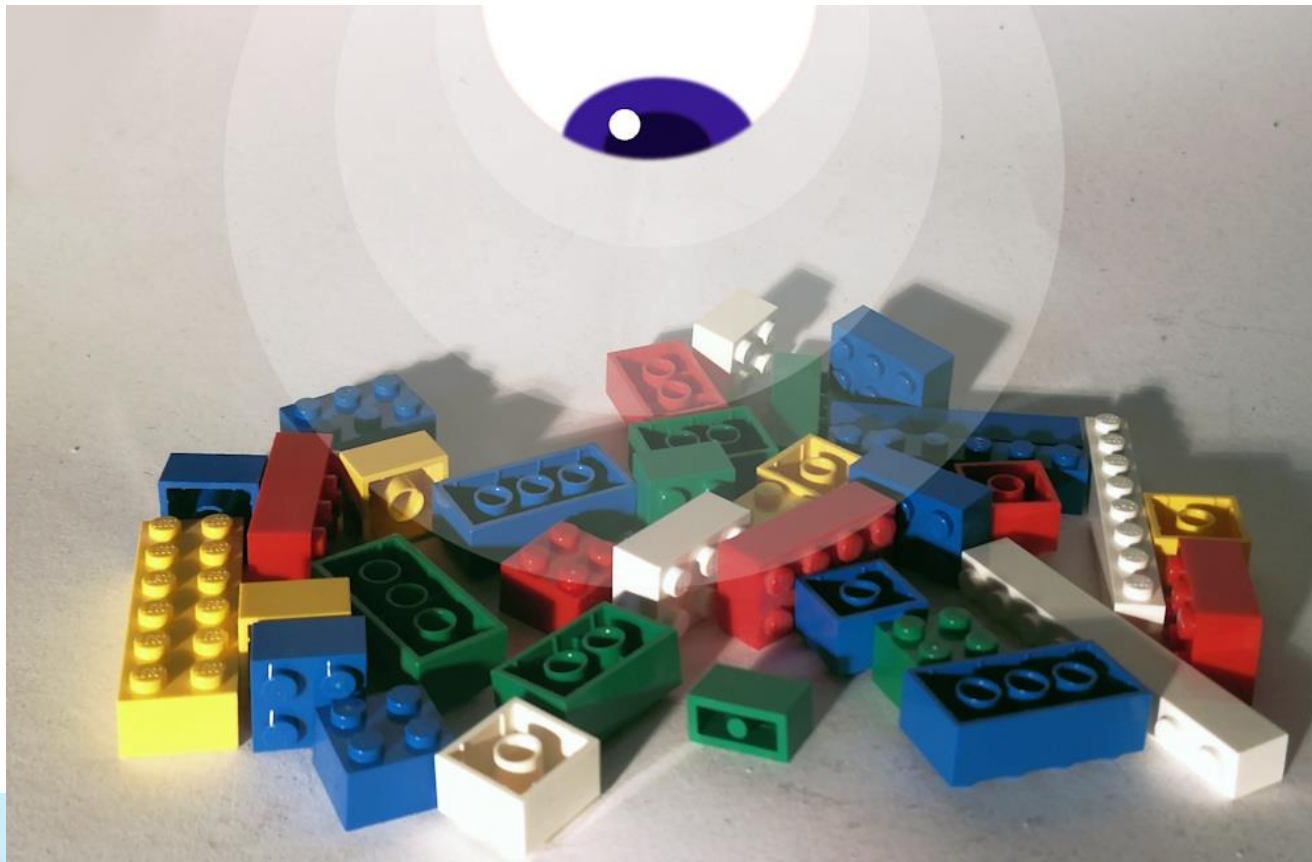
- Onko jokin asia kestävä muutos eikä päivänperho?
 - Onko se jonkin trendin terve jatkumo?
 - Onko siinä ristiriitoja isoihin trendeihin? (Esim. tietosuoja, kyberturvallisuus, energian käyttö...)



SYSTEMIEN PIIRTEET



Millaisia maailmamme systeemit ovat? Miten ne ovat muuttuneet ajan myötä konkreettisesti ja käsitteellisesti – ja miksi?



Näkemykset kehitettävistä systeemeistä 1/2

- Tietotekniikan kehitys nähdään joskus teknisenä asiana, mutta organisaatiomielessä on ymmärretty niiden sosiotekninen luonne: toimintajärjestelmän kokonaisuus on kehittämisen kohde.
- Nyt systeeminen ajattelu laajenee ja aletaan myös ymmärtää, että ne ovat luonteeltaan erilaisia: tuttuja ja turvallisia, kompleksisia, kaoottisia jne... (ks. Cynefin https://en.wikipedia.org/wiki/Cynefin_framework)
- Aiemmin systeemien piirteitä kuvattiin vain suoraviivaisilla attribuuteilla (konseptin uutuus, koko jne...)
 - Eli ajatus oli, että perusteiltaan asiat ovat samanlaisia, ne ovat vain joiltain osin erilaisia....

Näkemykset kehitettävistä systeemeistä 2/2

- Toisaalta ATK-systeemit eivät ole omassa hallinnassa olevia monoliitteja, vaan osa systeemien systeemejä, ekosysteemejä ja myös poliittisia systeemejä.
- *Tässä muutoksessa on kehitystä, mutta edelleen haasteita. Se liittyy systeemien konseptisuunnittelun haasteisiin, mikä on tuotekehityksessä olennaista.*
- *=> Kokonaisuuksien ymmärtäminen: mistä asiat koostuvat, mikä on erityistä, mikä taas ei.*



Laatukäsitys rikastuu

- Laatu on alue, jolle tulee aina uusia laatutekijöitä:
 - Toiminnallisuus -> luotettavuus -> suorituskyky -> käytettävyys -> energiataloudellisuus -> saavutettavuus -> kyberturvallisuus -> vastuullisuus
- *Tämä on inkrementaalista kehitystä, jolle ei tule loppua. Firmat ovat tietysti kyvykkyydessään aina pari pykälää perässä siitä, mikä olisi suotavaa.*
- *Mikä on seuraava laadun ja vaatimusten alue?*
- *=> Pitää olla tietoinen, fiksu, kehittyvä*



Kompleksisuus kasvaa

- Enemmän toiminnallisuutta -> enemmän komponentteja -> enemmän vuorovaikutuksia -> enemmän monimutkaisuutta
- *...ja siitä johtuen enemmän bugeja, enemmän tietoturvariskejä, vaikeammin ymmärrettäviä, vaikeammin testattavia, vaikeammin muutettavia ja kehitettäviä*
- *=> Tätä vastaan on taisteltava kaikin keinoin – keinoja on onneksi monia (palvelumuotoilu, skoupin hallinta, hypen hallinta, arkkitehtuuri...)*



ATK kantaa merkityksiä

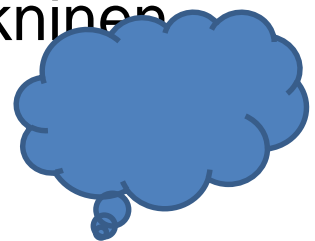
- Kaikki ei ole aina rationaalista.
- Se, millaista tekniikkaa organisaatiolla on, voi olla osin
 - Tarkkaan harkittu valinta
 - Toimialan default-ratkaisu, kulttuurinen meemi
 - Uusi hype
 - Mielikuvien ohjaava valinta
- Menestyvän merkkejä ovat eri aikoina olleet (kärjistyksenä) esim.:
 - Tietokone sinänsä -> IBM -> PC:t -> SAP, Oracle -> pilvi -> tekoäly
- Eettisen edelläkävijän merkkejä puolestaan (kärjistys)
 - Tietokone sinänsä -> Mac -> avoimen lähdekoodin ratkaisut -> palvelumuotoilu ja saavutettavuus
- => *Hankkijoiden kannattaa miettiä mielihalujaan ja toimittajien tehdä syvällistä "semanttista muotoilua" – brändäys, kilpailuettujen muotoilu...*

Tietotyöntekijä ja merkityksellisyys

- Tietotyöntekijä on systeemin inhimillinen elementti, josta ei tässä setissä puhuta kauheasti muuten kuin linkittyen merkityksellisyyteen.
- Tietotyöntekijän huomioon otto:
 - Jaksaminen huolellisena -> työergonomia -> työhyvinvointi -> työn houkuttelevuus -> työn merkityksellisyys
- Nämä kaikki ovat asioita, joihin teknisellä systeemillä on vaikutuksena.
- *Sanoilla on voimaa. Mitä viestii työnteko systeemeillä, joiden nimi on vaikka Sirpa, Sisu, Kilta, Odo?*

Suhde käyttäjiin syvenee 1/3

- Käyttäjäsuhteen taustalla on ihmiskuvan kehitys:
 - Suurten tarinoiden aikana käyttäjät ovat "standardi-ihmisiä", joilla on standarditarpeet -> persoonat ja stereotyypit nostavat erilaisuutta esille -> ihmiskuva rikastuu ja tarpeet voivat olla mitä tahansa (syvä asiakasymmärrys!)
- Tämän rinnalla kulkee ergonomisen perinteen kehitys:
 - Tietokone-ergonomia -> lomakeohjeet -> alustojen ohjeet -> sosiotekninen ajattelu pohjoismaisessa systeemityössä -> käytettävyysajattelu (kontekstuaalisuus) -> käyttäjäkokemus -> palvelumuotoilu -> saavutettavuus -> häiriökäytön hallinta
- => *Syvällinen ymmärrys on valttia*



Suhde käyttäjiin syvenee 2/3

- Toisaalta systeemeistä on usein haluttu tehdä sosiaalisia tai niihin on liitetty sosiaalisia apusysteemejä, esim. käyttäjien keskustelu- ja tukipalsta.
 - Silloin tulee käyttäjien turvallisuuskin yhdeksi varmistamisen kohteeksi (ja tietosuoja tietenkin).
- => *Käyttäjät kohtaavat aina kokonaistuotteen!*

Suhde käyttäjiin syvenee 3/3

- *Valmiudet ja tasot käyttäjälähtöisyydessä rikastuvat koko ajan, samoin myös pakolliset vaatimukset julkisissa ja yksityisen sektorinkin palveluissa (eli saavutettavuusvaatimukset). Tämä on kumulatiivista rikastavaa kehitystä.*
- *Mutta joskus ollaan näennäisellä tasolla. Esim. käyttöliittymäsuunnittelun kutsuminen UX:ksi ei tee siitä sitä.*
- *Ja miksi saavutettavuus ja muut koetaan firmoille pakoksi eikä mahdollisuudeksi kohdata asiakkaat paremmin?*
- *=> Käyttäjäkokemus on syytä kohdata aidosti, mahdollisuutena, ei kielipelillä*

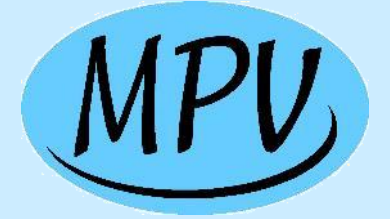


Käyttöliittymätyypit

- Tekstipäätteet ja numerovalikot -> Alasvetovalikot ja hiiri -> selainkäyttöliittymät -> portaalit ja intranet -> graafikot töihin -> rajoitetut mobiiliversiot ja appit -> reaktiivisuus -> saavutettavuus
 - Puheohjaus, eleohjaus yms. tietojärjestelmissä marginaalisia.
- *Käyttöliittymissä näkyy tekninen ja toiminnallinen konvergenssi eri ympäristöissä.*
- *=> Mobile first -filosofia alkaa yleistyä. Vihdoinkin. Designereille haaste.*
- *=> Saavutettavuus on uusi käytettävyys. Nyt siihen suhtaudutaan pakkona, vaikka sen pointti on asiakkaiden ja käyttäjien saavuttaminen! (Eikä vain palvelun toimintojen ja datan saavuttaminen käyttäjille)*



Tehdäänkö ihan oma tietojärjestelmä, räätälöity vaiko vain konffataan valmis



- Koko historian ajan on mietitty, että kannattaako ostaa valmis järjestelmä ja sovittaa toiminta siihen, vaiko toisinpäin - kuten organisaatioihmiset arvelevat järkeväksi.
- Kyse on tietysti aina rahasta ja vaivasta, ja räätälöinti tuo aina omia haasteitaan:
 - Ylläpito, turvallisuus, laatu...
- Kuitenkin, ihan puhtaalta paperilta ei enää kannata kehittää mitään, koska avoimen lähdekoodin systeemejä riittää aina pohjaksi.
- Joskus niistä tehdään toimialan kirjastoja – kuten kuntasektorilla.
- *Avoin lähdekoodi muuttaa koko ajan strategioita.*



Vai kaikki palveluna?

- Omistamisen kehitys:
 - Oma ohjelmisto omalla koneella -> oma tai liisattu hostauspalvelussa -> systeemi pilvessä -> infra ja sovellukset palveluna
- *Kun asiat arkipäiväistyvät, onko softa kuin sähköä: ERP-pitilaus tehdään simppelisti palveluntarjoajalta ja vaihdetaan käden käänteessä toiseen, datat ja kaikki siirtyen? (WordPress-saittien yms. simppelien siirto nettihotellista toiseen onnistuu jo melkein helposti.)*

Avoimuus oletukseksi

- Avoimen lähdekoodin idea alkoi elää vasta 1990-luvulla, vaikka epäviralliseen nörttikulttuuriin kuului paljon sitä ennen koodin luonnollinen jakaminen.
- Avoimen lähdekoodin systeemejä epäiltiin kovasti vielä 2000-luvun alussa, mutta ei enää. Nyt alkaa suljettu systeemi olla epäilyttävä poikkeus.
- Avoimuudella onkin monia etuja: joskus ilmaisuus, mahdollisuus tutkia muuttaa ja korjata, voi vaikka tehdä oman forkin, jos tuntuu siltä.
- Samaa avoimuuden ideaa sovelletaan yhä enemmän vaikkapa kuntien datassa - samaan aikaan, kun data nähdään kauppatavarana, standardeissa ja myös laitteisto-disaineissa.
- *Tässä asiassa on tapahtunut ajattelumallien ja käytäntöjen flippaus.*

Suunnittelunormit ja standardit

- Normit ovat formaaleja tai kulttuurin luomia yhteisiä uskomuksia ja sopimuksia.
- Säädökset ja standardit ovat niiden ilmentymiä.
- Tekniset sellaisia ovat hienoja: niiden avulla töpseli sopii pistorasiaan, komponentit ovat vaihdettavia ja ymmärrettäviä, ylläpito helpottuu jne...
- Kun tekniikka monimuotoistuu ja laatumalli rikastuu normien määrä kasvaa.
 - Perus datatekniikka (lukujen esittäminen) -> tietomallit (ml. relaatiomalli) -> tietoturvallisuus -> tietosuoja -> saavutettavuus
- => *Normit pitää tuntea*
- => *Hyvistä normeista kannattaa pakon sijaan löytää positiivinen etu. Esim. tietosuojadirektiivinen näkeminen asiakaslähtöisen etiikan kehikkona. Ja vastuullisuus kilpailuetuna.*

Missä dataa prosessoidaan?

- Tietokoneet ja niiden laskenta voidaan sijoittaa moneen paikkaan:
 - Osaston tai kaupungin jaetut koneet -> konesalit ja tyhvät päätteet -> henkilökohtaiset tietokoneet -> tietokoneet päätteinä -> pilvi -> reunalaskenta
- Ja monella tavalla:
 - "Raudalla" -> virtuaalikoneet -> kontit
- *Paljon huojuntaa edestakaisin. Muodit näyttävät arkkitehtuurin mahdollisuuksia. Kunkin ajan tarpeet suuntaavat kehitystä.*

Sovellusten määrä ja kokonaisarkkitehtuuri

- Sovelluksia tarpeeseen -> IT:n hallinta -> järjestelmien aika -> systeemien määrän räjähdys edelleen (satoja per organisaatio) -> monoliitit -> kokonaisarkkitehtuuri
- Ohjelmistorobotiikka seuraa samaa kaavaa, koska sitä suositellaankin kehitettävän bottom-up ja kokeillen.
- => *Kokonaisarkkitehtuuri on ATK:n hyödyntäjillä välttämätön asia.*



ATK-älyn muodot muuttuvat

- Ohjelmallinen äly on tietysti enimmäkseen proseduraalista, algoritmista. Logiikkaohjelmointi toi alkua tekoälyyn (Prolog) ja sumea logiikka teki siitä joustavampaa.
- Nyt ollaan koneoppimisen dominanssissa.
- Uutta "talvea" ei kenties tule, mutta Watsonkin on kai konkurssitilassa.
- Uutta on se, että systeemit eivät ole miljoonahankintoja, vaan niitä voi hankkia edullisesti. Haaste onkin data.
- *Älykkyys lisääntyy ja uudistuu, mutta kohtaa aina haasteita.*
- *=> Järjen käyttö järjessä on noussut paljon esille. Koneoppiminen ei ole itseisarvo, vaan pitää miettiä missä sitä käytetään ja pitää osatakin sen käyttöä.*

Datan käyttö 1/2

- ATK:n alussa käytettiin dataa niin vähän kuin mahdollista, koska pikkuruiset keskusmuistit, massamuistit ja prosessoritehot rajoittivat kapasiteettia.
 - Tietoliikenne ei vielä paljon haitannut, koska sitä ei juuri ollut, korkeintaan tyhmille päätteille.
- Sen jälkeen datan käyttö onkin kasvanut eksponentiaalisesti sekä yksittäisissä sovelluksissa että tietenkin yhteensä.
 - Big Data -ajattelu analytiikkaa varten sai kaikki miettimään datan keruun maksimointia ja tekoäly tuplasi paineet siihen.
 - Nyt on taas oivallettu, että paljon huonoa dataa ei auta, datan käsittely maksaa ja on ympäristöhaitta, ja iso osa datasta voi olla eettisesti epäilyttävää - markkinointivakoilua.
 - Ja pienten älylaitteiden pitää toimia edelleen hyvin säästäväisesti ilman patterihuolia.

Datan käyttö 2/2

- Nyt ollaan tietopohjaisia ja dataperustaisia kaikessa!
- Ja data tietysti on tekoälyn polttoainetta.
 - Sillä tehdään operaatioita, mutta myös tulkitaan tilanteita systemien kehittämisen lähtökohdaksi.
- Datan kehittyminen:
 - Yksittäisen prosessin informaatio -> tietomallit (relaatiomalli apuna) -> rikas ajatus datasta: missä muodossa tahansa, sumeaa, monin tavoin linkittynyt
- Haastenäkökulma:
 - Bisnesinformaation esitys -> varastointi -> datojen yhdistäminen -> tietoturvallisuus -> tietosuoja -> analytiikka ja koneoppiminen
- => *Haasteet kasvavat koko ajan. Osaamistarveavaruus laaja*



Datan arvo oivallettu

- Data oli ATK:n alussa yleensä neutraali, välttämätön väline asioiden aikaansaamiseksi.
- Vasta 2000-luvulla alettiin ymmärtää varsinkin ihmisten datan arvo markkinoinnissa - ja myös kerätä sitä.
- Vastaava ilmiö näkyy erilaisten prosessien tuottamasta datasta, jolle on käyttöä tuotekehityksessä, logistisissa systeemeissä yms. ja varsinkin tekoälyn opettaminen varsinkin tarvitsee todella paljon dataa
- => *Nyt pitää joka tilanteessa miettiä ja sopia datan omistajuus ja arvo.*



Kyberturvallisuus 1/2

- Tietoturvallisuuden maailma oli helpohkoa ennen nettiaikaa, kun tietojärjestelmiä pyöriteltiin suljetuissa spesiaaliverkoissa.
 - Henkilökohtaisille tietokoneille toki tarjottiin viruksia ladattavissa ohjelmissa.
- Kun taloussystemit, tehtaot, hallinto ja kaikki muu menivät nettiin, maailma muuttui.
- Nyt päästiin systemaattisen kyberrikollisuuden ja kybersodan aikaan
- Mihin turvallisuusajattelu vaikuttaa?
 - Käyttöroolit ja -oikeudet -> systeemien suojaus -> turvallinen suunnittelu -> systeemin pärjääminen hyökkäyksissä -> toipuminen hyökkäyksistä



Kyberturvallisuus 2/2

- Tässä on kyseessä portaittainen transformaatio lintukodosta sodankäyntiin.
- => *Jokainen systeemi on turvallisuuskriittinen ja sitä pitää suhtautua sellaisena*
- => *Jokainen toiminnallisuus on lähtökohtaisesti turvallisuuskriittinen, olipa se sellaiseksi luokiteltu tai edes "vaatimukseksi"*

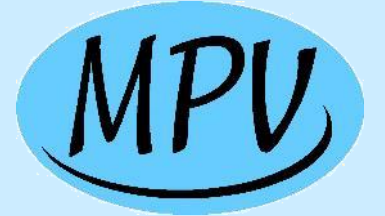


Päivitystahti

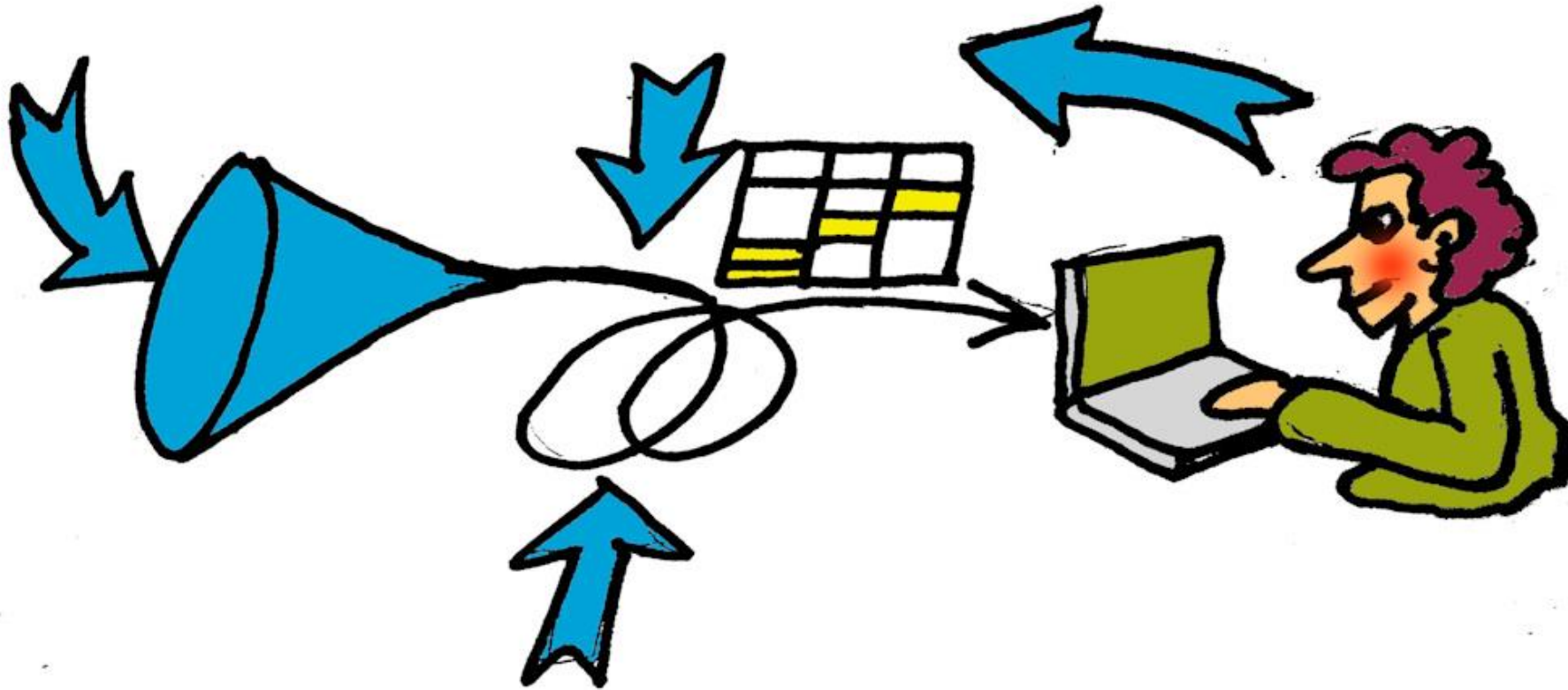
- Asioiden pitää kehittyä, kun bisnes muuttuu, palvelut muuttuvat jne...
- Siksi tietojärjestelmiä pitää päivittää
 - Kerran tai pari vuodessa-> jatkuva päivittäminen
- Pieni muutos on ongelmaton.
- Päivitys harvaksen tekee muutokset näkyviksi, ne tiedetään kouluttaa, ohjeet ja dokumentit päivittää... Mutta kirotaan, kun tarvittavia asioita pitää odottaa
- Jatkuva muutos on niin pientä, että siihen on hankala reagoida.
- Tieto asioista vanhenee hallitsemattomasti eikä aina tiedetä mikä versio toimii milläkin tavalla. Tuttu juttu Linuxista... Joka asiaan on 100 tietoa, joista 70 vanhentunutta
- => *Hyvä trendi, mutta vaatii myös hallintaa*



TYÖTAVAT SYSTEMEIMIEN KEHITTÄMISESSÄ



Miten niitä systeemejä saadaan aikaan? Millaisilla prosesseilla, millaisella yhteistyöllä jne...



Prosessimallit 1/3

- ATK:n alussa ei ollut kehittämisprosessien menetelmämalleja. Vähitellen päästiin niiden maailmaan vastineena kehittämisen systemaattisuuden vaatimuksille:
 - Vesiputousmalli -> EVO -> RUP -> XP -> Scrum -> SAFe
- Prosessimallit ovat muoteja ja brändejä, jotka vaihtuvat ja joista haetaan tarttuvaa arvoa. Aina ne heijastavat “maailman henkeä”.
- Agilen mallit olivat vastapainoa prosessikeskeisyydelle, johon palataan taas SAFe:ssa. Syklistä sykkimistä... kevyestä raskaaseen... kevyeseen... jne...



Prosessimallit 2/3

- "Välihuomio: Agileen kantapää"
 - Joskus muutos vie vähäksi aikaa huonompaan suuntaan.
 - "Agile-liikettä" on kritisoitu kahden asian unohtamisesta: arkkitehtuuri ja käyttäjäkeskeisyys
 - Asioiden keventäminen vei ohjelmointisentriseen ajatteluun johtuen "liikkeen" avainhenkilödemografiasta
 - => *Pitää itse muistaa pitää huolta, että asiat tehdään fiksusti ihan muodeista riippumatta*

Prosessimallit 3/3

- Mutta systeemytyössä on aina pykälää ylemmän tason runko, jossa:
 - Selvitetään nykytila
 - Suunnitellaan uuden systeemin peruseriaatteet
 - Tehdään systeemytyötä
 - Otetaan uutta systeemiä käyttöön
- ...Tilanteesta riippuen enemmän tai vähemmän ketterästi tai isommissa tai pienemmissä paloissa. (Sen päällä vielä hankintamalli.)
- Nykytilan selvitys muuttuu:
 - Työtavat ja käytettävät välineet -> liiketoimintaprosessin rikas tunnistaminen
-> kokonaistoiminnan mallinnus + datavarastot + datalähteet
- *Siksi Scrumit yms. ovat vain isomman kuvion runko.*

Ohjelmointikielet

- Olennaisimmat tietojärjestelmien kannalta:
 - RPG -> COBOL -> SQL -> C [nettiajan alku] -> Perl -> Java -> PHP -> C# -> JavaScript...
- Tietojärjestelmä voi tietysti olla hajautettu kokonaisuus, joka kokoaa dataa monesta lähteestä, joissa on käytössä muita kieliä
 - Python, R, C++...
- *Kieliä tulee ja menee. JS on hypetetty ja kaikkialla, mutta PHP ykkönen nettipohjaisissa systeemeissä. Kielet eivät ole vain syntaktinen valinta, vaan tuovat mukanaan frameworkit, kirjastorepositoryt, riippuvuudet, tietoriskit ja kehittäjäyhteisöt.*
- *SQL:n rooli? Piilossa tietokannoissa, kuulostaa vanhentuneelta, mutta voisi tehdä paljon enemmän #lähelläDatavarastoja*

Arkkitehtuurityö 1/2

- Arkkitehtuuria ei ensimmäisissä systeemeissä tarvinnut juuri miettiä, koska se kumpusi tietokoneen käyttötavasta.
- Arkkitehtuurin merkitys kasvoi ajan mittaan systeemien koon ja kompleksisuuden kasvaessa.
- Jossain kohtaa taisi tulle arkkitehtuuriväsymys, kun sovelluskehitys ja systeemityö olivat pelkkää UML-jumppaa.
- Agilen myötä siirryttiinkin ohjelmointikeskeiseen kulttuuriin, jossa arkkitehtuuri syntyy kuin luonnostaan... Nyt siitä on kenties opittu.
 - Arkkitehtuurityö ml. arkkitehtuurin arviointiauttaa välttämään virheitä ja ylilyöntejä

Tapoja perussuunnitteluun

- Perussuunnitteluun, konseptiin jne... on monta reittiä:
 - Bisneksen, prosessin tarve -> valmisohjelma, johon toiminta sovitetaan -> kokeilukulttuuri -> palvelumuotoilu
- => *Olellista on löytää oikea tapa tilanteeseen*



Arkkitehtuurityö 2/2

- Arkkitehtuurityö näkyy konsulttipuheessa dokumentointina, mutta sen pitäisi tietysti olla suunnittelua, jolle on uusia ajureita:
 - Turvallisuus.
 - Muutettavuus, kun asioita tapahtuu ja pitää reagoida vaihtamalla, siirtämällä...
 - Riippuvuudet (vrt. NPM lisää niitä, konteilla hallitaan...)
 - Yms...
- *Arkkitehtuurityön maailma kertoo oppimisesta, tasapainon hakemisesta ja myös hallinnan tarpeesta, kun haasteet kasvavat.*

Tieto onnistumisesta

- Prosesseissa on iso asia tiedon saaminen kehittämisen onnistumisesta.
- Empiirinen palaute käytöstä:
 - Vuosittainen julkistus & palaute pääkäyttäjän kautta -> käytettävyytestaus
-> pienet julkaisut usein -> käytön monitorointi
- Ja palaute suunnitelmiin
 - Asiakkaan osallistuminen vaatimusmäärittelyyn -> asiakkaan osallistuminen tiiviisti kehittämiseen
- Sama pätee sisäiseen testaukseen:
 - Harvat julkistukset testaustiimille -> jatkuva testaus kehittämissä
- *Palautteen nopea saaminen onkin yksi kaiken läpäisevä trendi.*

Rytminen reflektio

- Toimijoiden itsereflektio on tärkeää projektien riskienhallinnalle ja kehittymiselle.
- Sen muutosta:
 - Itsearviointi ja asiakaspalaute projektin päätteeksi ja ohjausryhmäkatselmointi -> arviointi (Lessons Learned) projektin vaiheiden päätteeksi -> agilen retropektiivit sprinttien päätteeksi ja joustava sidosryhmäviestintä -> toiminnan tiivis reflektointi coachin kanssa
- *Reflektion tiivistyminen on heijastusta ympäröivästä maailmasta ja johtamisen maailman kehityksestä.*

Testaus ja laadunvarmistus

- Testauksen kehittyminen heijastelee olennaisten laatuominaisuuksien kasvua:
 - Toiminnallisuus -> käytettävyys -> kuormituksen kesto / suorituskyky -> tietoturvallisuus -> saavutettavuus
- Ja työtyyli työelämän muutosta – organisoituminen, automaatio, älyn käyttö...
 - Ohjelmoija testaa & debuggaa -> testitapausten ajo käsin testausryhmissä -> testiautomaatio -> tutkiva testaus
- *Kumpikin kehityskaari jatkuu. Diversiteetti ei vielä näy (paitsi automaatio ja tutkiva testaus –yhdistelmässä), mutta hyvä testaus ja laadunvarmistus on monimuotoista. Sitä tuovat analyttiset menetelmät – käytettävyyden ja saavutettavuuden arviointi, arkkitehtuurin arviointi jne...*
- *=> Pitää varmistaa kokonaislaatua. Sen varmistaminen, mitä voi testata automaattilla, ei riitä*



Testausaika

- Kun uusi versio tai päivitys on "valmis" -> sisäisille buildeille suunnitelman mukaan -> jatkuva testaus: testiautomaatio ja yksikkötestaus
- Käytettävyys- ja suorituskykytestaus edelleen liian myöhään, jos ollenkaan
- *Trendinä tehdä testausta aikaisemmin ja jatkuvasti*

Johtaminen

- Klassiseen työtoimintaan on kuulunut ajatus systemaattisesta top-down -työnjaosta ja päätäntävallasta. 2000-luvulla on noussut ajatus itseohjautuvuudesta tiimitasolla, jolloin johtamisen rooli siirtyy fasilitointiin ja valmentamiseen.
- *Tämä muutos on jatkuvaa kulttuurin muutosta, joka heijastelee yhteiskunnan kehittymistä.*

Prosessinormit ja prosessi-standardit

- Prosessistandardeihin on usein suhtauduttu varauksella, koska organisaatioiden pitäisi aina löytää itselleen parhaat työtavat.
- ATK:n alussa ei prosessistandardeja ollut. ISO- ja IEEE nousivat myöhemmin tärkeiksi.
- Menetelmät, kuten Scrum, ovat olleet defacto-standardeja. Suomessa julkishallinnon VAHTI-ohjeistus on ollut ja on tärkeä.
- ISO 9001 (ja 90003) on tietysti metatason standardi, joka esittää varsinaisille käytännöille vaatimuksia.
- Kypsyysmallit ovat pois muodista, hyvästä syystä
- => *Normeilla ja prosessistandardeilla on omat paikkansa, mutta niitä on sovellettava fiksusti (ja valittava pohjaksi oikeat). Jos toiminta on fiksua, sen normisovitus yleensä onnistuu*

Yhteistyö organisaatiossa

- Kaikissa asioissa on samanlaisia mekanismeja:
 - Aloitus pienesti -> ammattikunnan synty -> organisatorinen eriyttäminen -> uusi yhteistyön kaipuu ml. Tiimitys
- *DevOps ja DevSecOps ovat tällaista: Kiinteytystä eriyttämisen jälkeen.*

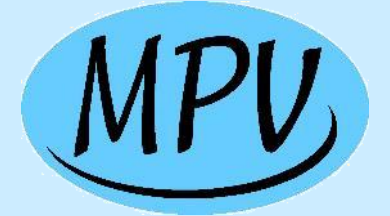
Lähellä vai etänä

- Vuonna 2022 on triviaalia nähdä, että lähityöstä on siirrytty paikkariippumattomaan työhön, eli yleensä ns. etätyöhön.
- “Turvalliset” tietoverkot ovat tietysti tämän mahdollistajia, eli ennen nettiäikää etätyö olisi ollut hankalaa.
- Mutta pandemia laukaisi kollektiivisiakin psyykkisiä esteitä.
- *Tässä on kyse työn rakenteiden uudelleenajattelusta.*

Automaatiota eri paikoissa

- Systemien kehittämisessä on automaatio apuna rutiineissa.
- Systemien rakentaminen
 - Kokoaminen käsityönä & kopiointi käsin tuotantoon -> buildaus-skriptit -> CI-palvelin -> continuous deployment -> kaiken määrittely ohjelmallisesti
- (Toiminnallinen) testaus
 - Testaus käsin ad hoc -> testitapausten ajo käsin -> yksikkötestauksen puoliautomaatio -> rakentamiseen integroitu testiautomaatio
- Tieto käytöstä
 - Havainnointi -> kaiken lokitus
- *Tämä on inkrementaalista muutosta, pyrkimys maksimointiin*
- *=> Fiksu automatisoi rutiinit*

SYSTEMIEN TEKIJÄT



Entäpä ihmiset? Niiden ajattelusta ja osaamisesta kaikki lähtee.

*Tehtävä Asema Rooli Ammatti Titteli Koulutus Ydinosaaminen T-malli
Pätevyys Sertifikaatit Osaamisen taso Ikä Sukupuoli Kulttuuritausta
Maailmankuva Etiikka Vastuullisuus Tapa toimia Nopeus
Temperamentti Tiimiosaaminen Ahkeruus ja huolellisuus
Käyttäytyminen Henkilöbrändi Ammattibrändi Viestintäosaaminen
Innovointikyky Osaamisen kehittäminen Sitoutuminen Urakehitys Yms.*

Erikoistuminen

- Ensimmäiset ohjelmoijat tekivät kaiken.
 - Sitten tuli laajeneva specialisaation aika ja valtava osaamisalueiden fragmentaatio.
 - Palattiin full-stack-ajatteluun - mutta välinekeskeisesti. Mutta aina syntyy uusia osaamisalueita. "Full-stack" kattaa vain tekniikkaa, ei asiakaskokemusta tai saavutettavuutta.
- Samaan aikaan työelämän väitetään kaipaavan moniosaajia tai T-mallista osaamista.
- *Tämä on kuplivaa kaaosta. Tekniikan kupla on vaarallinen.*

Työryhmät, tiimit

- ATK alkoi yksilöiden työstä.
- Jo 1970-luvulla määritettiin iso joukko tiukkoja rooleja, jotka täytti joku henkilö.
- Työ oli ketjutettu roolien läpi.
- Ryhmiä oli tietysti käytössä: kehittäjillä ja testaajilla omansa.
- Agilen myötä siirryttiin tiimeihin ja rajat paljolti sulivat. Tiimin piti yhdessä hoitaa kaikki.
- Mutta! Osaamisalueiden täydentyessä ei-ohjelmoivilla alueilla (saavutettavuus, palvelumuotoilu, tietosuoja, tietoturvasuus) palataan taas väliaikaisesti työhön osallistuvien experttien aikaan.
- *Tämä on syklistä siirtymistä tyylistä toiseen.*

Ideaali organisaatio ja eri aikojen huolenaiheet

- Ahkera ja huolellinen -> systemaattinen, prosessilähtöinen, laadukas -> tehokas (tehdas) -> asiakaslähtöinen -> ketterä ja nopea -> luova -> resilientti
- *Organisaatioon liittyvät ihanteet muuttuvat ajan myötä, mutta osa entisistä ideaaleista säilyy aina.*
- *Mikä on tärkeää, riippuu kontekstista.*



Resurssointi

- ATK:n alussa kaikki tehtiin itse. Sitten alihankittiin asioiden kehittämistä ja tuotantoa.
- Ulkoistus ulkomaille oli muotia, josta taas palataan.
 - Taustalla ajattelu kustannuksista ja ydinosamisesta. Ehkä organisaatio voi olla vain virtuaalinen?
- *Tämä on kiertokulkua, johon vaikuttavat monet tekijät. Kuitenkin, nopeat turvalliset tietoverkot muuttavat perinteisiä raja-aitoja.*



Kehittäjät ja sukupuoliterveys

- ATK:n alkuvuosina ohjelmoijat olivat useimmiten naisia, koska ohjelmointi koettiin vain vähän kykyjä vaativaksi konttorityöksi.
- Sitten kulttuuri muuttui ja stereotypia kehittämistiimistä on joukko kovin samanlaisia miehiä. Kaikilla vielä firman T-paita päällä.
- *Vähitellen syntyy vastavoimia (Mimmit koodaa) ja kulttuuri muuttuu. Korkea aika. Ollaan matkalla IT-työelämän tervehtymiseen.*
- => *Terve organisaatio on nykymaailmassa kestävä*

Osaaminen 1/2

- ATK:n alkuaikoina piti pioneerien hakea koulutusta valmistajien kursseilta.
- Pian syntyi yliopistotason koulutusta ja myöhemmin laajaa opistotason opetusta.
- Koulutusmahdollisuudet ovat moninaisia ja nykyään kursseja katsellaan netistä.
- Osaamistarpeet myös fragmentoituvat, kuten kaikki muukin.
- Yritykset ovat aikalaila luopuneet työntekijöiden koulutuksesta, poikkeuksena rekrytointikoulutus kampanjaluonteisesti (esim. pilvi- tai testiautomaatio-osaajat)
 - Valmistajien kurssit -> koulutusjärjestelmä -> yksityiset koulutuspalvelut -> verkkokurssit -> rekrykoulutus

Osaaminen 2/2

- (Coaching on yleistynyt työtapa-asioissa työnohjauksen muotona, eli on vähän eri juttu)
- Harrastus on aina ollut toinen reitti osaamiseen.
 - Nykyään avoimen lähdekoodin ohjelmien käyttö mahdollistaa välineidenkin oppimisen
 - Ja avoimen lähdekoodin ohjelmien kehittäminen mahdollistaa osaamisen ja yhteistyökyvyn näyttämisen
- Osaaminen linkittyy yleiseen yhteiskunnalliseen ajatukseen jatkuvasta oppimisesta
- *Osaamisen hankkiminen siirtyy enemmän yksilön vastuulle, mutta myös mahdollisuudet siihen kasvavat.*

Osaamisen ja ongelmanratkaisun jatkuva tuki

- Työssä tarvitaan aina jatkuvaa tukea.
- ATK:n alussa moni oli aika yksin.
- Organisaatioiden kasvaessa syntyi kollegoita ja yhdistyksiä.
 - Tietoverkoissa oli Usenet / Sfnets News tärkeä ennen nykyajan foorumeja, kuten Stack Overflow.
 - Kevyt vertaistuki -> Kasvavien organisaatioiden kollegiaalinen tuki -> Usenet News -> Verkkofoorumit -> Valmistajien yhteisöfoorumit
 - Yleisyhdistykset -> teemayhdistykset -> fragmentoituminen nippeleiksi...
- *Yhteisöjen merkitys on valtava.*

PARI LINKKIÄ

- Julkaisuluettelon teema "tulevaisuuden tuotteiden ja käyttöliittymien kehittäminen"

https://mattivuori.net/julkaisuluettelo/#tulevaisuuden_tuotteiden_ja_kayttoliittymien_kehittaminen

- Testauksen kehityskaaria in English:

https://mattivuori.net/julkaisuluettelo/liitteet/trajectories_in_testing.pdf