Jukkapekka Lalu

Ohjelmistoautomaation lisääminen ohjelmistoprosessiin ja ylläpitoon

Insinööri

Tieto- ja viestintätekniikka

Kevät 2022

KAMK • University of Applied Sciences

Tiivistelmä

Tekijä(t): Lalu Jukkapekka

Työn nimi: Ohjelmistoautomaation lisääminen ohjelmistoprosessiin ja ylläpitoon

Tutkintonimike: Insinööri (AMK), tieto- ja viestintätekniikka

Asiasanat: prosessi, ohjelmistoautomaatio, ohjelmistoprosessi, sidosryhmät

Opinnäytetyön toimeksiantajana toimi Raute Oyj. Työn tavoitteena oli luoda prosessi, jolla voidaan lisätä ohjelmistoautomaatiota ohjelmistoprosessiin. Ohjelmistoprosessia käsiteltiin terminä ja sen kehittämisen hyötyjä. Ohjelmistoautomaatiossa tutustuttiin automaation rakenteeseen ja kannattavuuden arvioimiseen sekä automaation todelliseen arvoon. Kannattavuuden arvioinnissa perehdyttiin hyötyihin ja haittoihin. Automaation todellisessa arvossa pohdittiin, millaisista asioista se koostuu.

Tässä työssä käsiteltiin sidosryhmiä terminä sekä osana ohjelmistokehitysprojektia. Erilaisia sidosryhmätyyppejä ja sidosryhmien analysoinnin prosessia tutkittiin. Sidosryhmien analysoinnissa perehdyttiin prosessiin ja siitä saataviin hyötyihin.

Työssä luotiin prosessi ohjelmistoautomaation lisäämisestä. Prosessi sisälsi neljä syklistä vaihetta. Nykyisen tilanteen arvioinnissa tutkittiin ohjelman nykyinen tilanne ja mahdollisuudet automaatioille. Ohjelmistoautomaation esisuunnittelussa perehdyttiin tarkemmin mahdollisiin automaatioihin. Kannattavuuden arvioinnissa päätettiin automaation toteuttamisesta. Automaation toteuttamisessa toteutettiin automaatio suunnitelmien mukaisesti.

Prosessin toimivuutta arvioitiin esimerkkiprojektin avulla. Esimerkkiprojektin aikana prosessista ei löytynyt isompia puutteita. Prosessin eri vaiheet hoitivat oman osuutensa, ja kokonaisuudessaan prosessi toimi niin kuin se oli suunniteltu. Täten prosessi todettiin varsin toimivaksi arvioinnissa. Sidosryhmien analysointi huomattiin asiaksi, jota voisi mahdollisesti korostaa prosessissa entistä enemmän. Sidosryhmien analysointia voidaan korostaa lisäämällä nykyisen tilanteen arviointiin oma maininta siitä. Täten sidosryhmien analysointi olisi osa nykyisen tilanteen arviointiin kuuluvia toimenpiteitä.

KAMK • University of Applied Sciences

Abstract

Author(s): Lalu Jukkapekka

Title of the Publication: Addition of Software Automation to Software Process and Maintenance

Degree Title: Bachelor of Engineering, Information and Communication Technology

Keywords: process, software automation, software process, stakeholders

This Bachelor's thesis was commissioned by Raute Corporation. The primary goal of this thesis was to create a process which can be used to add software automation to a software process. The software process and its benefits of development were explored. At software automation, its structure, profitability assessment and real value of automation were inspected. At profitability assessment, possible benefits or negatives were examined. At real value of automation, the parts which affect the real value were inspected.

Stakeholders and their part in the software development project were explored. Different stakeholder types and the stakeholders' process analysis were examined. At analyzing the stakeholders', the process and its benefits were inspected.

This work creates a process of adding software automation to a software process. The process included four cyclic phases. At current situation evaluation, the current situation of the program and potential automation opportunities were explored. At software automation pre-design, potential automations were inspected more specifically. At profitability assessment, decisions about implementing automation were made. At automation implementation, the automation was implemented based on designs.

The functionality of the process was evaluated with an example project. During the example project, the process worked well and no big problems were found. All different phases of the process worked as planned. This is why the process was found to be working as intended during its evaluation. Analyzing the stakeholders was found to be worth highlighting even more in the process. Analyzing stakeholders could be highlighted more with the addition of a mention of it during the current situation evaluation. This way, analyzing the stakeholders would be part of the measures in the current situation evaluation phase.

Sisällys

1 Johdanto

Tässä työssä tutkitaan, millaisella prosessilla voidaan kehittää ohjelmistoprosesseja ohjelmistoautomaation avulla ja käyttöönottaa ne huomoiden kaikki osapuolet. Työssä käydään läpi niin ohjelmistoprosessin kuin automaation teorioita yleisellä tasolla. Täten nämä aiheet tulevat tutuiksi ennen kuin ruvetaan käymään läpi varsinaista prosessia ja siihen liittyviä asioita.

Kappaleessa 2 tutustutaan, mikä on ohjelmistoprosessi ja mitä se käytännössä tarkoittaa. Tämän lisäksi käydään yleisesti läpi, miten niitä voidaan kehittää sekä minkä takia niiden kehittäminen on hyödyllistä.

Kappaleessa 3 käydään läpi ohjelmistoautomaatiota ja sen rakennetta. Kannattavuuteen liittyviä seikkoja ja miten automaation lisäämisen todellista arvoa voidaan määritellä.

Kappaleessa 4 tutkitaan mitä tarkoittaa termi sidosryhmä, millaisia sidosryhmätyyppejä on ja miten sidosryhmiä analysoidaan sekä miksi niiden analysointi on hyödyllistä.

Kappaleessa 5 aiheena on itse prosessi ohjelmistoautomaation lisäämisestä ohjelmistoprosessiin ja sen käyttöönotto. Kappaleessa myös pohditaan, miten prosessia voidaan kehittää entuudestaan.

Kappaleessa 6 tutustaan esimerkkiprojektiin, jossa on toteutetttu ja käyttöönotettu automaatiota kappaleen 5 prosessin mukaisesti. Tämän avulla saadaan esitettyä prosessin toimintaa ja hyötyjä käytännön esimerkin kautta.

Kappaleen 7 aiheena on prosessin arviointi, jossa arvioidaan kappaleessa 5 luodun prosessin toimivuutta.

Kappaleessa 8 käydään vielä yhteenvetona läpi työssä käydyt tärkeimmät asiat.

2 Ohjelmistoprosessi ja sen kehittäminen

Ohjelmistoprosessi on joukko toisiinsa liittyviä toimintoja, jotka johtavat ohjelmiston tuotantoon. Nämä toimet voivat sisältää ohjelmiston kehittämisen alusta alkean tai olemassa olevan järjestelmän muokkaamisen. Ohjelmiston eritelmä, ohjelmiston suunnittelu ja toteutus, ohjelmiston varmennus ja validointi sekä ohjelmiston evoluutio ovat toimintoja, jotka jokaisen ohjelmistoprosessin täytyy sisältää. Ohjelmiston eritelmä määrittää ohjelmiston päätoiminnot ja niitä ympäröivät rajoitukset. Ohjelmiston suunnittelu ja toteutus pitää sisällään ohjelmiston suunnitelun ja sen ohjelmoimisen. Ohjelmiston varmennuksen ja validoinnin mukaan ohjelmisto on eritelmän ja asiakkaan tarpeiden mukainen. Ohjelmiston evoluution mukaan ohjelmistoa muokataan vastaamaan asiakkaiden ja markkinoiden vaatimia muutoksia. [1.]

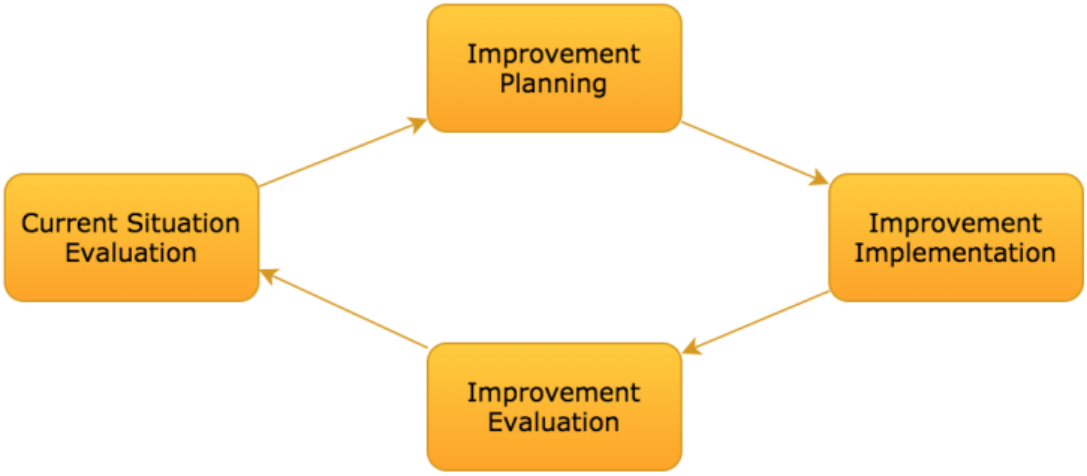
Ohjelmistoprosessi on monimutkainen asia ja se perustuu päätösten tekemiseen. Yhtä ylitse muiden olevaa ihanteellista prosessia ei ole olemassa, minkä takia useimmat organisaatiot ovat kehittäneet omat ohjelmistoprosessinsa. Esimerkiksi kriittisten järjestelmien parissa työskentelevällä organisaatiolla on hyvin jäsennelty prosessi, kun taas liiketoimintajärjestelmissä, joissa vaatimukset muuttuvat nopeasti, vähemmän muodollinen ja enemmän joustava prosessi on todennäköisesti tehokkaampi vaihtoehto. [1.]

Kaikki suuret pyrkimykset alkavat kysymyksestä. Tässä tapauksessa kysymys on sama jokaisella yrityksellä huolimatta heidän toimialastaan tai heidän asemastaan. Kysymys on, miten voimme olla entistäkin parempia siinä, mitä teemme. Tämän kysymyksen yhteydessä ohjelmistoprosessien kehittäminen astuu mukaan kuvaan. Jokaisella toimialalla, joka käyttää tietokoneita, on käyttöä ohjelmistoprosessien kehittämiselle. [2.]

Ohjelmistoprosessien kehittäminen määritellään tehtävien, työkalujen ja tekniikoiden sarjaksi, jolla suunnitellaan ja toteutetaan parannustoimia tiettyjen tavoitteiden saavuttamiseksi, kuten tuotantonopeuden lisääminen, korkeamman tuotelaadun saavuttaminen tai kustannusten alentaminen. [3.]

On olemassa paljon todisteita ohjelmistoprosessien kehittämisen arvosta, mikäli ne ovat toteutettu onnistuneesti. [3.] Tästä huolimatta ohjelmistoprosessien kehittämiseen liittyy myös negatiivisia asioita, minkä takia sitä ei aina tehdä, vaikka sille olisi tarvetta. Ohjelmistoprosessien kehittämiseen käytetty aika ja raha on pois jostain muualta, minkä takia lyhyellä aika välillä katsottuna sitä ei aina koeta kovinkaan korkealle erilaisten asioiden tärkeysjärjestyksessä.

Ohjelmistoprosessin kehittäminen koostuu pääasiassa neljästä syklisestä vaiheesta, kuten alla olevassa kuvassa 1 näkyy, kun taas nämä vaiheet voidaan jakaa useisiin vaiheisiin käytetyn menetelmän ja tekniikoiden mukaan. [3.]



Kuva 1. Ohjelmistoprosessin kehittämisen neljä syklistä vaihetta [3].

Current Situation Evaluation eli nykyisen tilanteen arviointi on prosessin aloitusvaihe. Tämän vaiheen pääasiallinen tavoite on arvioida ohjelmistoprosessin nykytilannetta saamalla esiin sidosryhmien vaatimukset, analysoimalla nykyiset ongelmat ja suoritteet sekä tunnistamalla ohjelmistoprosessin tehottomuudet. [3.]

Improvement Planning eli parannuksien suunnittelu on vaihe, jossa nykytilanteen arvioinnin perusteella suunnitellaan tarvittavat parannukset ja laitetaan parannukset tärkeysjärjestykseen [3]. Suunnittelu ja asioiden tärkeysjäjestykseen laittaminen, helpottaa asioiden tekemistä teoria tasolta käytäntöön.

Improvement Implementation eli parannusten implementointi tarkoittaa suunnitelman mukaisten parannusten tekemistä. Täten parannukset saadaan teoria tasolta käytäntöön. [3.]

Improvement Evaluation eli parannusten arviointi on prosessin viimeinen vaihe, jossa arvioidaan parannusten todellista arvoa [3]. Parannusten arvioinnin perusteella, pystytään tekemään johtopäätöksiä parannusten kannattavuudesta.

3 Ohjelmistoautomaatio

Ohjelmistoautomaatio tarkoittaa jonkun ohjelman vaiheen tai prosessin muuttamista toimimaan itsenäisesti ilman, että käyttäjän tarvitsee erikseen tehdä asioita juuri siinä ohjelman vaiheessa. Hyvänä käytännön esimerkkinä voidaan käyttää autojen tuulilasin pyyhkijöitä. Tuulilasin pyyhkijöillä on kaksi tilaa on- ja off-tila sekä muutamia eri nopeusasetuksia pyyhkijöiden ollessa päällä eli on-tilassa. Pyyhkijöiden ollessa on-tilassa ne liikkuvat itsekseen asetuksen mukaisesti, mikäli auton tuulilasin pyyhkijöitä ei olisi automatisoitu toimimaan nopeusasetusten mukaan. Tarvitsisi kuljettajan aina väännellä pyyhkijöiden vipua edestakaisin pyyhkiäkseen vettä pois auton tuulilasista. Tästä päästään myös siihen, miksi ohjelmistoautomaatiota ylipäänsä tehdään ja miksi se on hyödyllistä. Mietitään tilannetta, jossa tuulilasin pyyhkijöiden toimintaa ei olisi automatisoitu nopeusasetuksen mukaan, vaan kuljettajan tarvitsisi aina kääntää vipua, kun hän haluaa käyttää niitä ja olisi kova vesisade. Tuulilasin pyyhkijöiden manuaalisesti toimiminen vesisateella ei estäisi ajamista, mutta jopa kokeneelle ja hyvälle kuljettajalle tämä voisi aiheuttaa ongelmia, koska pyyhkijöiden käyttöön käytetty aika ja keskittyminen ovat pois itse siitä tärkeimmästä eli ajamisesta ja tämä taas voi pahimmassa tapauksessa aiheuttaa onnettomuuksia.

Lyhyesti ja yleisesti selitettynä ohjelmistoautomaatio virtaviivaistaa järjestelmää poistamalla tarvittavaa ihmisen osuutta toimiakseen. Tämä vähentää virheitä, nopeuttaa toimitusta, parantaa laatua, minimoi kustannuksia ja yksinkertaistaa prosessia [4].

3.1 Automaation kannattavuus

Saadakseen parempaa kuvaa, miksi jonkun asian automatisointi on kannattavaa, vastataan kysymykseen, mitä meidän pitäisi automatisoida. Tähän voidaan vastata vastaamalla toiseen kysymykseen, joka on, mikä ero on marmorin veistämisellä ja kivien rikkomisella. Tässä tapauksessa kiviä murskataan ja marmori kuvaveistetään vasaran kanssa. Työkalut ja väline ovat samat, mutta ero on mielen sitoutumisessa. Kivien murskaaminen on mielikuvituksetonta ja toistuvaa työtä. Kivien veistäminen vaatii ihmisen luovuutta ja kekseliäisyyttä. Annetaan koneiden hoitaa yksinkertainen kivien murskaaminen samalla, kun luotamme taiteilijoiden luomaan uusia ja upeita veistoksia. Vastaus kysymykseen, mitä meidän pitäisi automatisoida on, että automatisoidaan asiat, jotka ovat toistuvia, tylsiä, mielikuvituksettomia ja ei mielenkiintoisia,

koska näiden asioiden tekeminen polttaa ihmisiä loppuun. Tämän lisäksi koneet voivat hoitaa tämän tyyppiset tehtävät kaikin puolin paremmin kuin ihminen. [5.]

Automaation kannattavuus määritellään loppuen lopuksi hyödyistä, joita se voi tuottaa, sillä oletuksella, että siitä aiheutuvat haitat eivät ole liian isot. Mikäli aiheutuvia haittoja on paljon, pitää pohtia, onko saatavat hyödyt tärkeämpiä tai pystytäänkö haittoja estämään muulla tavalla.

3.1.1 Hyödyt

Automaation hyödyt
Tehokkuuden parantuminen
Parempi tuottavuus
Vähemmän kuluja
Työvoiman parempi käyttö
Parempi turvallisuus
Parempi tuotannon laatu
Parempi työympäristö
Vähemmän virheitä

Kuva 2. Automaation hyötyjä.

Yllä kuvassa 2 on listattu hyötyjä, joita voidaan saada automaatiosta. Kuvan lista on koostettu lähteistä 6 ja 7. Automaation hoitaessa aikaa vieviä yksitoikkoisia, toistuvia ja väsyttäviä työtehtäviä, vapauttaa se työntekijöille aikaa keskittyä tärkeämpiin ja mielenkiintoisempiin työtehtäviin [6]. Automaation tapana on myös hoitaa työtehtävät nopeammin, laadukkaammin, turvallisemmin, tehokkaammin ja yleisesti ottaen paremmin kuin ennen. Työtehtävien hoituessa automaation avulla avaa se yritykselle mahdollisuuden keskittää työvoimansa muihin työtehtäviin ja lisätä yhtäaikaisesti tehtyjen työtehtävien lukumäärää tai vähentää kuluja tarvitsemalla kaiken kaikkiaan vähemmän työvoimaa hoitaakseen samat työtehtävät kuin ennen.

3.1.2 Haitat

Automaation haitat
Työpaikkojen mahdollinen vähentyminen
Investointikustannukset
Riippuvuus tekniikasta
Ohjelman joustavuuden väheneminen
Arvaamattomat kustannukset
Huono asiakaspalvelu
Vaadittava taitojen päivittäminen

Kuva 3. Automaation haittoja.

Kuten useammissa muissakin elämän asioissa niin myöskään automaation lisääminen ei ole pelkkää juhlaa. Yllä olevassa kuvassa 3 on listattu haittoja, joita automaatio voi aiheuttaa. Kuvan lista on koostettu lähteistä 6 ja 7.

Automaation lisääntyminen muokkaa työmarkkinoita, koska jotkut työt tullaan korvaamaan automaatiolla [6]. Toisaalta samaan aikaan kehitetään uusia töitä, jotka vaativat työvoimalta parempia taitoja kuin ennen [6]. Yleensä yritykset, jotka lisäävät ohjelmistoautomaatiota pystyvät luomaan työntekijöilleen uusia töitä automaatiolla korvattujen tehtävien tilalle [7]. Tämä johtuu automaation tuomasta tehokkuuden lisääntymisestä, jonka takia yritys pystyy työskentelemään useamman projektin kanssa ja palvelemaan useampia asiakkaita yhtäaikaisesti [7].

Automaation lisääminen ei myöskään tapahdu täysin tyhjästä ja ilmaiseksi, vaan siihen sisältyy investointikustannuksia, kuten työntekijöitä, aikaa ja rahaa. Tämän lisäksi automaation lisäämisestä voi aiheutua arvaamattomia kustannuksia, kuten automaation tutkimus- ja kehityskustannukset, ennaltaehkäisevät huoltotoimenpiteet ja työvoiman kouluttaminen toimimaan automaation kanssa [6]. Ihmisten kanssa suoraan yhteydessä olevien palvelujen automatisointi voi aiheuttaa ikäviä tilanteita, koska tietokoneen kanssa ei ole mahdollista ruveta neuvottelemaan mitä tehdään vaan se tekee asiat sen mukaan, miten se on ohjelmoitu. Tämän takia automaation lisääminen asiakaspalvelutilanteisiin voi huonontaa niitä parantamisen sijaan. Automaatiosta voi myös aiheutua lisäkustannuksia ylläpitoon.

Ohjelmista tulee vähemmän joustavia automaation lisäämisen myötä, mikä voi aiheuttaa enemmän työtä ja ongelmia päivitysten tekemisen kanssa. Riippuvuus tekniikasta ja teknologiasta

ovat haittoja, jotka aiheutuvat automaation lisääntymisestä. Asioiden toiminnallisuuden ollessa riippuvainen yhdestä asiasta tuo tämä mukanaan omia huonoja puolia, vaikka kaikki huonot puolet eivät välttämättä ole täysin selviä nykyhetkellä, voi niitä ilmestyä lisää mitä enemmän automaatiota lisätään.

3.2 Automaation rakenne

On yksi asia tunnistaa automaation arvo ja toinen toteuttaa se. Suunnittelutyön ja automaation toteuttaminen ei tarvitse olla monimutkaista ja mystistä. Kyse on vain siitä, että tiedetään, mitä halutaan tehdä sekä tunnistetaan, mitä voidaan tehdä ja selvitetään, mitä työkaluja tarvitaan rattaiden liikkeelle panemiseksi. Alla kerrottujen vaiheiden muodostaman rakenteen avulla pystytään automatisoimaan mitä vain. [8.]

Aivan ensimmäiseksi pitää määritellä tavoitteet ja ymmärtää ne kunnolla. Tämä auttaa järjestelemään asiat tärkeysjärjestykseen tilanteen mukaan. Yleisesti oletetaan, että automaation tavoite on aina parantaa tehokkuutta, mutta se ei ole totta. Tavoitteita voi muun muassa olla tuotteen tai palvelun arvon parantaminen, prosessien tekeminen entistä ennalta arvattavimmiksi tai oikeanlaisiksi, asiakaskokemuksen parantaminen tai toistuvuuden ja ärsyttävyyden vähentäminen työntekijöiltä. Alapuolella olevassa kuvassa 4 on esimerkki tavoitteen määrittelystä, jossa on myös listattu yleisimpiä syitä, miksi juuri tietty tavoite on tarpeen. [8.]

Goal	Common issue(s)
Deliver better customer outcomes	Process is error prone, unpredictable, or slow
Improve employee experience	Too many repetitive tasks, takes up too much employee time
Increase efficiency	Process is too expensive, takes too long to complete, or not scalable

Kuva 4. Tavoitteen määrittely [8].

Rakenteen toinen vaihe on tunnistaa prosessi, jota ollaan automatisoimassa ja asettaa rajat sille. Helpoin tapa asettaa rajat ja estää automaation skaalauksen laajeneminen on tunnistaa, mikä aloittaa prosessin, jota ollaan automatisoimassa ja tunnistaa sen prosessin lopputulos. [8.]

Seuraavana vaiheena tunnistetaan kaikki tehtävät, jotka täytyy tehdä suorittaakseen prosessin ja tunnistaa, mitkä näistä tehtävistä voidaan automatisoida. Tehtäviä, jotka vaativat ihmisen valvontaa tai tekemiä päätöksiä, ei voi automatisoida. Alla kuvassa 5 on selitetty yleisimpiä tehtävien tyyppejä, joita voidaan automatisoida. [8.]

Quality	Description
Repetitive	Involves the same steps or inputs each time
Frequent	High volume task that must be completed multiple times in an hour, day, or week
Recurring	Tasks must be completed according to a regular schedule
Dependent	Triggered by another event or change in status
Simple	Does not require complex information, decision making, or problem solving
Predictable	Planned element of a workflow or process, occurs in every instance of the process
Collaborative	Requires action or input from multiple stakeholders

Kuva 5. Yleisimpiä tehtävien tyyppejä, joita voidaan automatisoida [8].

Rakenteen viimeisessä vaiheessa suunnitellaan, toteutetaan ja lopuksi testataan automaatio omien kykyjen ja käytössä olevien työkalujen avulla parhaalla mahdollisella tavalla. [8.]

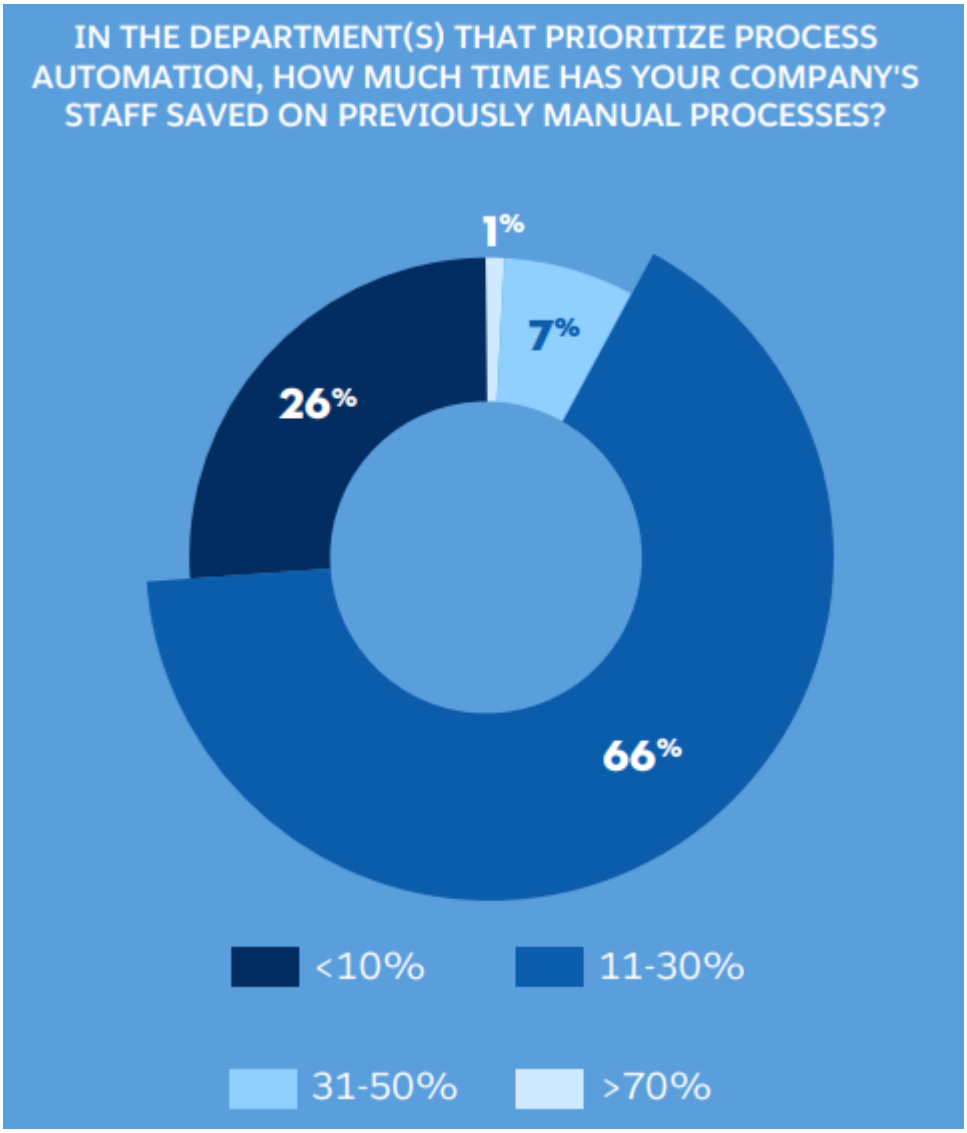
3.3 Automaation todellinen arvo

Ohjelmistoautomaation todellisen arvon arvioiminen ennen automaation toteutusta on elintärkeää yrityksen tehdessä päätöstä, kannattaako automaatio toteuttaa vai ei. Tämän arvioiminen on kuitenkin erittäin hankalaa. Automaation arvoon vaikuttavat automaatiosta

saatavat hyödyt suhteessa siitä aiheutuviin haittoihin, joita on aiemmin avattu kappaleissa 3.1.1 ja 3.1.2.

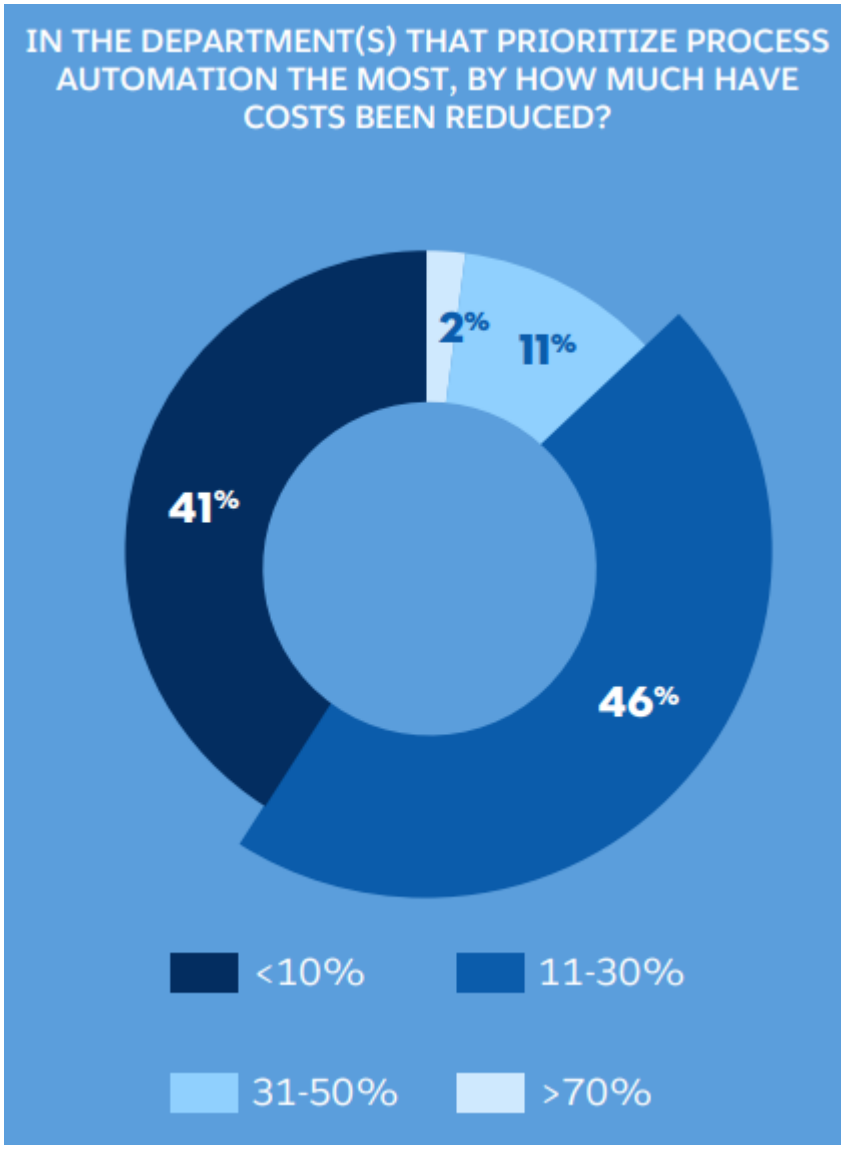
Hyötyjen arvoon vaikuttaa monet tekijät. Arvoon vaikuttavia tekijöitä ovat muun muassa automaation nopeus suorittamaan prosessi suhteessa ihmiseen, automaation kestävyys suorittaa prosessia uudelleen ja uudelleen, automaation vapauttama aika keskittyä johonkin muuhun, automaation lisäämä turvallisuus ja kulujen vähentyminen. Nopeus ja kestävyys voidaan arvioida omina arvoinaan, mutta ne myös vaikuttavat automaation avulla vapautuvaan aikaan. Automaation avulla vapautuvalle ajalle voi antaa ainakin jonkinlaisen arvon arvioimalla, kuinka paljon sitä aikaa vapautuu. Tosin sen ajan täysin todellista arvoa on hankala arvioida, koska sitä arvioitaessa ensinnäkin pitäisi tietää, mihin se aika käytetään ja toisaalta osata arvioida sen ajan arvo.

Yhdysvaltalainen ohjelmistoyritys nimeltä Salesforce järjesti syksyllä 2020 kyselyn 100 maailmanlaajuiselle IT- ja suunnittelujohtajalle, jossa selvitettiin muun muassa, millaisia vaikutuksia automaation käytöllä on ollut heidän yrityksiinsä. Tutkimalla tämän kyselyn tuloksia saadaan jonkinlaisia kuvia siitä, millaisista arvoista puhutaan automaation vapauttaman ajan ja kulujen vähenemisen suhteen. [9.]



Kuva 6. Kyselyn lopputulos automaation vapauttamasta ajasta [9].

Kuten kuvassa 6 näkyy 66 % vastanneista arvioi, että heidän yrityksensä työntekijät säästävät 11–30 % ajasta, jonka he ovat aikaisemmin käyttäneet nykyään automaation korvaamiin asioihin. Tutkimuksen perusteella voidaan todeta 74 % vastanneista, säästää yli 10 % aikaisemmin manuaalisesti tehdyistä asioista nykyään automaation avulla.



Kuva 7. Kyselyn lopputulos kulujen vähenemisestä [9].

Kuvan 7 perusteella voidaan todeta 46 % vastanneista, säästävät kuluissa 11–30 % automaation avulla. Kaiken kaikkiaan 59 % vastanneista arvioi kulujen vähentyneen yli 10 % automaation takia.

Aikaisemmin mainitulle mahdolliselle turvallisuuden lisääntymiselle automaation takia on myös hankala antaa jotain tiettyä taloudellista arvoa, mutta jokainen työtapaturmasta aiheutuva sairaslomapäivä maksaa yritykselle aikaa ja rahaa. Täten automaation yhtään tuoma lisäturvallisuus voi lisätä automaation tuomien hyötyjen arvoa ilman, että sitä edes kunnolla huomaa. Kuten tästäkin asiasta voidaan huomata, niin automaation todellisen arvon määrittäminen hyötyjen suhteen ei ole kovinkaan helppoa etenkään, jos yritetään arvioida, onko automaation tekeminen järkevää vai ei ennen kuin sitä on tehty.

Haittojen arvioinnin suhteen asiat helpottuvat ainakin jossakin määrin. Tosin tämä vaatii arviointia ennen hyvän tutkimustyön automaation toteuttamisen ja käytettävien ohjelmistojen sekä laitteiden suhteen. Käytettäviin ohjelmistoihin tai laitteisiin voi tulla päivityksiä, jotka vaativat muutoksia automaation toimivuuden suhteen. Täten lisätä automaatioon vaadittuja kustannuksia.

Hyötyjen ja haittojen arvoon vaikuttaa myös ohjelman käytön yleisyys sekä käyttäjien lukumäärä. Ohjelman yleinen käyttö tai käyttäjien iso lukumäärä, voi moninkertaistaa automaatiosta saatavat hyödyt tai aiheutuvat haitat kokonaisuudessaan. Yleisesti ottaen pohtiessa, onko jonkun tietyn automaation toteuttaminen järkevää vai ei, pitää päätös tehdä arvioimalla parhaalla mahdollisella tavalla siitä saatavia kokonaishyötyjä suhteessa potentiaalisiin kokonaishaittoihin.

4 Sidosryhmät

Termi sidosryhmä viittaa ihmisiin tai ryhmiin, joihin ohjelmistokehitysprojekti vaikuttaa. Sidosryhmiä on organisaatiossa sekä sen ulkopuolella. He voivat olla ohjelman varsinaisia käyttäjiä tai prosessi voi yksinkertaisesti vaikuttaa heihin. Tämän takia heillä on oma kiinnostuksensa lopputuotteeseen. Sidosryhmien palaute kertoo yritykselle, millaista ohjelmistoa tarvitaan, ehdottaa ideoita ominaisuuksista tai ongelmista, joita tarvitsee ratkaista. Sidosryhmät ovat parhaimmassa asemassa tarjoamaan erityistä panosta tarpeisiinsa omalla tasollaan. He tietävät, mikä toimii ja mikä ei toimi heidän työnkulussaan. Lisäksi luokkansa etujen edustajina heillä on käsitys kaikista ainutlaatuisista tarpeista, jotka voivat olla ristiriidassa muiden sidosryhmien kanssa. Tämän tiedon hankkiminen varhaisessa vaiheessa kehitystä auttaa kehittäjiä löytämään kompromissin ennen kuin vakavia ongelmia ilmenee. [10.]

Sidosryhmien laiminlyönti on riskialtista. Ensinnäkin ilman heidän panostaan kehittäjät työskentelevät epätäydellisen vaatimusluettelon perusteella. Yllätystarpeita ilmaantuu varmasti kehityksen aikana. Nämä äkilliset lisäykset voivat hyvin nopeasti aiheuttaa projektin kasvamisen yli alkuperäisten rajojensa. Täten alkuperäiset aika- ja budjettivaatimukset joutuvat venymään uusien vaatimusten kattamiseksi. Näiden venyminen ei ole aina mahdollista. On paljon todennäköisempää, että joitain ominaisuuksia joudutaan leikkaamaan määräaikojen noudattamiseksi. Vaikka määräajat ja budjetit pysyisivät suunniteltujen rajojen sisällä, puuttuvilla osuuksilla on seurauksia. Joissakin tapauksissa ohjelmisto toimii juuri niin kuin johto sen halusi toimivan, mutta työntekijät eivät käytä sitä. Heillä saattaa jo olla tehokkaampia työkaluja tai uudessa ohjelmistossa ei ole heidän haluamiaan ominaisuuksia. Oli syy mikä tahansa, heidän innostuksensa puute johtaa hukkaan menneeseen sijoitukseen. [10.]

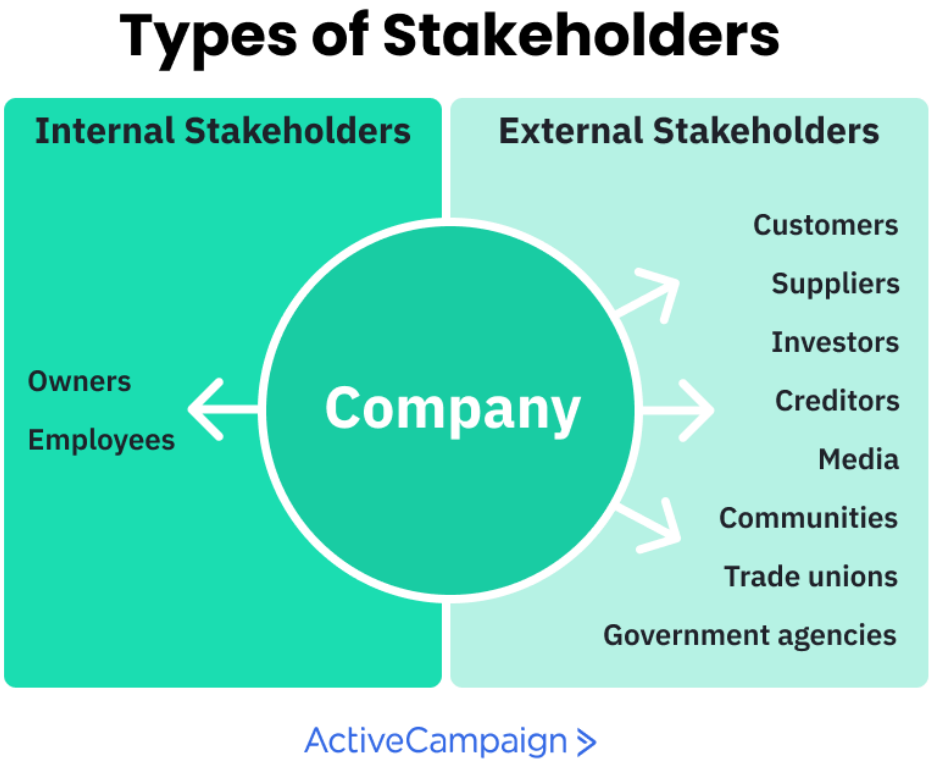
4.1 Sidosryhmätyypit

Jokainen sidosryhmätyyppi luokitellaan kolmella tavalla. Sisäinen tai ulkoinen, ensisijainen tai toissijainen ja suora tai epäsuora [11].

4.1.1 Sisäinen tai ulkoinen

Sisäiset sidosryhmät ovat sidosryhmiä, jotka ovat yrityksen sisällä. Nämä ovat sidosryhmiä, joihin hanke vaikuttaa suoraan [11]. Kuten kuvasta 8 voi nähdä, esimerkiksi työntekijät kuuluvat sisäisiin sidosryhmiin.

Ulkopuoliset sidosryhmät ovat niitä, jotka ovat kiinnostuneita yrityksen menestyksestä, mutta joilla ei ole suoraa yhteyttä organisaation projekteihin [11]. Kuvasta 8 voidaan ottaa esimerkiksi tavarantoimittaja, joka kuuluu ulkopuolisiin sidosryhmiin.



Kuva 8. Esimerkki sidosryhmätyyppien jakamisesta sisäisen ja ulkoisen välillä [11].

4.1.2 Ensisijainen tai toissijainen

Ensisijaiset sidosryhmät ovat eniten kiinnostuneita hankkeen lopputuloksesta, koska lopputulos vaikuttaa heihin suoraan. He osallistuvat aktiivisesti projektiin. Tälläisiä sidosryhmiä ovat muun muassa asiakkaat ja tiiminjohtajat. [11.]

Toissijaiset sidosryhmät auttavat myös projektien toteuttamisessa, mutta alemmalla, yleisellä tasolla. Tämäntyyppiset sidosryhmät auttavat hallinnollisissa prosesseissa, taloudellisessa ja juridisissa asioissa. [11.]

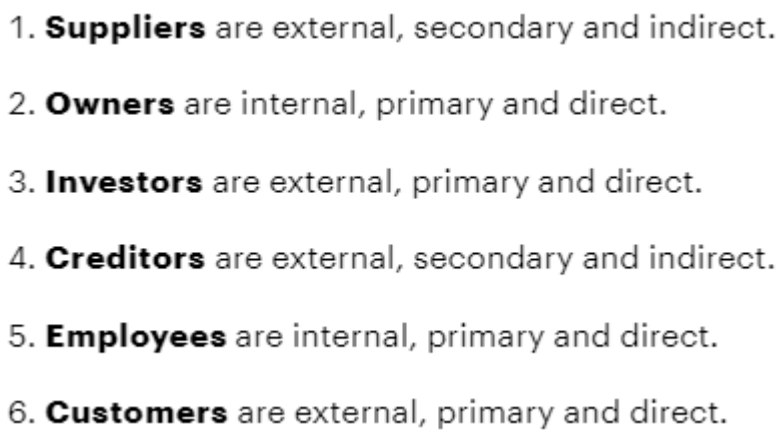
4.1.3 Suora tai epäsuora

Suorat sidosryhmät ovat mukana projektin päivittäisessä toiminnassa. Työntekijöitä voidaan pitää suorina sidosryhminä, sillä heidän päivittäiset tehtävänsä pyörivät yrityksen projektien ympärillä. [11.]

Epäsuorat sidosryhmät kiinnittävät huomiota valmiin projektin lopputulokseen eikä sen loppuun saattamiseen. Epäsuorat sidosryhmät ovat huolissaan sellaisista asioista kuin hinnoittelu, pakkaukset ja saatavuus. Asiakkaat ovat eräänlainen epäsuora sidosryhmä. [11.]

4.1.4 Sidosryhmien perusluokittelu

Sidosryhmien perusluokittelussa annetaan jokaiselle kolmelle luokitukselle arvo. Alapuolella olevan kuvan 9 esimerkin mukaan esimerkiksi employees eli työntekijät kuuluvat sisäisiin, ensisijaisiin ja suoriin sidosryhmiin, kun taas suppliers eli tavarantoimittajat kuuluvat ulkoisiin, toissijaisiin ja epäsuoriin sidosryhmiin.



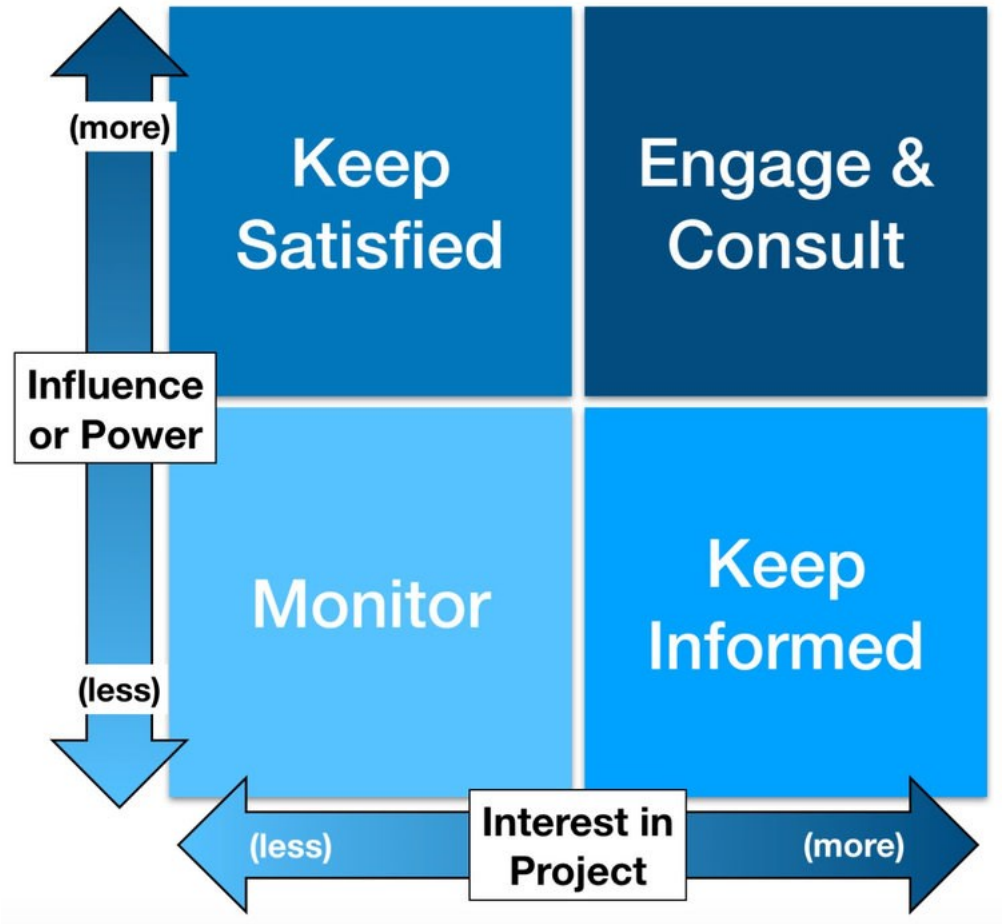
Kuva 9. Esimerkki Sidosryhmätyyppien perusluokittelusta [12].

4.2 Sidosryhmien analysointi

Sidosryhmien analysointi on prosessi, jossa eri sidosryhmiin kuuluvat ihmiset tunnistetaan ennen projektin alkamista. Prosessissa ryhmitellään ihmiset osallistumisen, kiinnostuksen ja projektiin vaikuttamisen mukaan ja määritetään, kuinka kukin sidosryhmä pidetään ajan tasalla asioista. [13.]

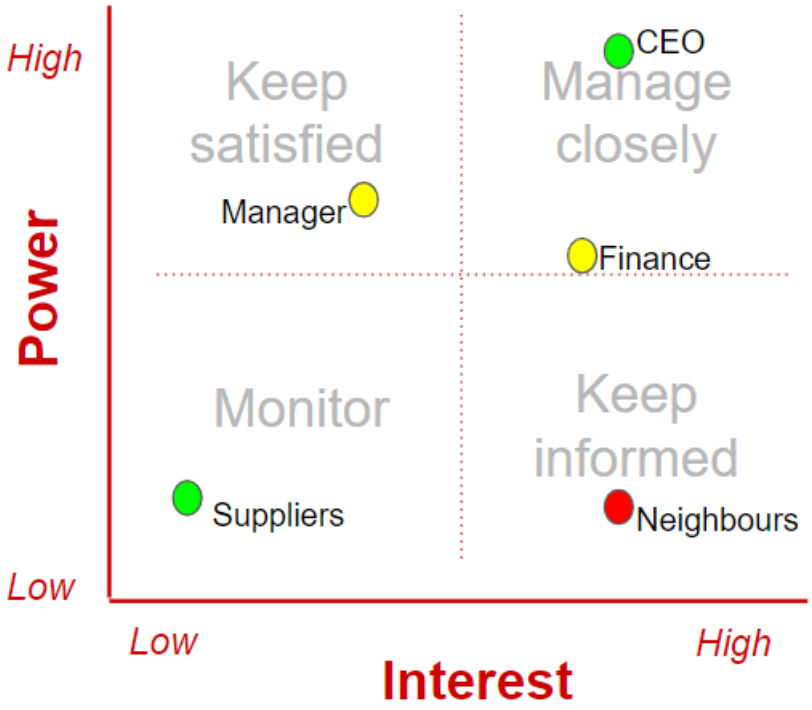
Sidosryhmien analysoinnin tarkoituksena on löytää projektin kannalta tärkeimmät sidosryhmät. Tämän tiedon avulla heidät voidaan ottaa huomioon jo suunnitteluvaiheessa ja täten ratkaista mahdolliset ongelmatilanteet hyvissä ajoin, sekä sidosryhmät ovat tietoisia projektin lopputuotteen tuomista muutoksista. [13.]

Analysointi aloitetaan listaamalla kaikki mahdolliset sidosryhmät. Tämän jälkeen kaikki sidosryhmät asetetaan kuvan 10 mukaiselle power–interest-ruudukolle sen mukaan, millainen vaikutus ja kiinnostus kullakin sidosryhmällä on projektiin nähden. [13.]



Kuva 10. Power-Interest ruudukko [14].

Kuvan 10 ruudukossa engage&consult-ruutuun sijoitetut sidosryhmät ovat niitä, jotka ovat elintärkeitä projektin onnistumisen kannalta, ja heidät kannattaa pitää tyytyväisinä projektin aikana sekä tietoisina projektin edistymisestä. Keep satisfied -ruudun sidosryhmät ovat sellaisia, joita pitää yrittää pitää tyytyväisinä, koska heillä on valta vaikuttaa projektin tekemiseen, vaikka heille ei ole merkittävää merkitystä, millainen on projektin lopputulos. Keep informed -ruudun sidosryhmillä ei ole valtaa vaikuttaa projektin tekemiseen, mutta he ovat kiinnostuneita projektin lopputuloksesta, esimerkiksi projektin lopputulos voi vaikuttaa heihin ja siksi olisi hyvä pitää heidät tietoisina projektista. Monitor-ruudussa olevilla sidosryhmillä ei ole valtaa vaikuttaa projektin tekemiseen eikä heille ole merkitystä, millainen projektin lopputulos on. Tosin heitä ei kannata jättää täysin huomioitta, vaan enemmänkin tarkkailla varmistaakseen, että heidän valta- ja kiinnostustasonsa projektin suhteen eivät olennaisesti muutu projektin tekemisen aikana. [14.]

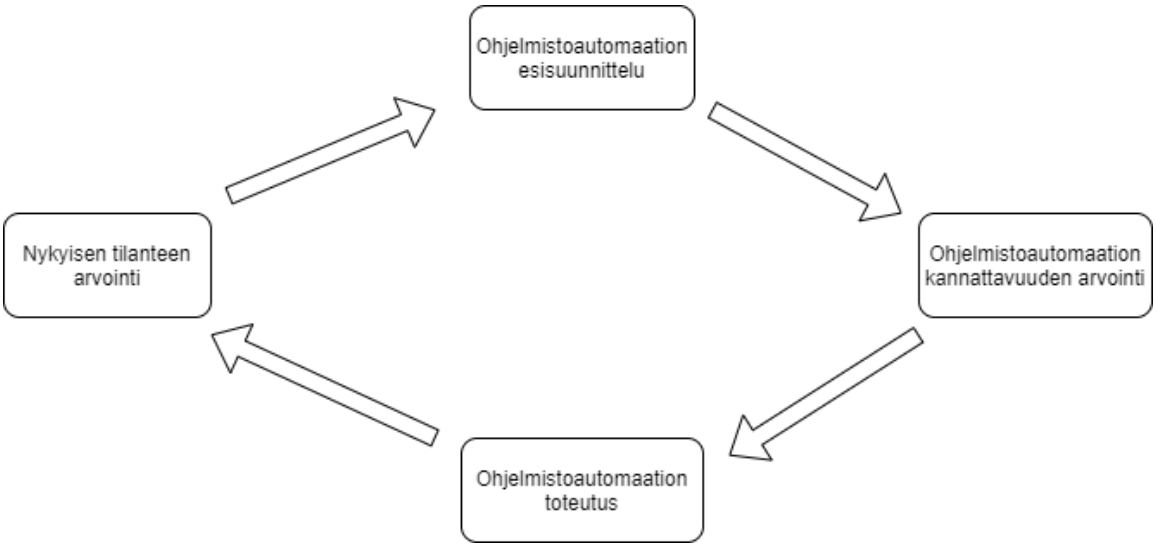


Kuva 11. Esimerkki power–interest-ruudukosta sidosryhmien kanssa [15].

Kuvassa 11 on esitetty esimerkki power–interest-ruudukosta sidosryhmien kanssa. Tilanne sidosryhmien suhteen voi muuttua projektin aikana ja siksi sidosryhmien analysointiin kannattaa palata suhteellisen tasaisin väliajoin projektin aikana.

5 Ohjelmistoautomaation lisääminen ja prosessin kehittäminen

Kun oli tutustuttu tarkemmin ohjelmistoprosessiin ja ohjelmistoautomaatioon, kehitettiin näiden tietojen perusteella prosessi ohjelmistoautomaation lisäämisestä ohjelmistoprosessiin ja ylläpitoon. Alla kuvassa 12 on esitetty prosessi vaiheittain.



Kuva 12. Prosessi ohjelmistoautomaation lisäämisestä.

Prosessi alkaa nykyisen tilanteen arvioinnilla. Tämän vaiheen tarkoituksena on tutkia ohjelman nykyinen tilanne ja samalla arvioida, millaisille ohjelmistoautomaatioille voisi olla tarvetta ohjelmassa. Lähtökohtaisesti sellaiset ohjelman vaiheet, joihin voidaan lisätä automaatiota ovat toistuvia, tylsiä, mielikuvituksettomia tai ei mielenkiintoisia käyttäjälle, kuten kappaleessa 3.1 todettiin. Näiden lisäksi potentiaalisia automatisoitavia asioita ovat aikaa vievät ja käyttäjän keskittymistä vaativat ohjelman vaiheet. Kuvassa 13 on esitetty nykyisen tilanteen arvioinnin sisältämiä asioita. Ohjelman nykyistä tilannetta lähdetään tutkimaan, jonkun ongelman takia. Ongelmaan pyritään löytämään ratkaisuja esimerkiksi automaation avulla. Tämän lisäksi ohjelmaan liittyvät sidosryhmät kannattaa selvittää tässä vaiheessa prosessia. Sidosryhmien kautta on mahdollista tulla esille ratkaisuja ongelmaan.

Nykyisen tilanteen arviointi
Ongelma
Ratkaisumahdollisuudet
Sidosryhmät

Kuva 13. Nykyisen tilanteen arvioinnin sisältämät asiat.

Nykyisen tilanteen arviointi -vaiheen jälkeen on ohjelmistoautomaation esisuunnittelu. Ohjelmistoautomaation esisuunnittelussa on tarkoituksena perehtyä paremmin ensimmäisessä vaiheessa esille tulleisiin ohjelmistoautomaatioihin, joiden toteuttamiselle olisi tarvetta.

Lista ohjelmistoautomaation esisuunnitelu vaiheen kysymyksistä
Miten automaatio toteutetaan?
Minne automaation lisäämisestä aiheutuvat muutokset tehdään?
Mihin asioihin muutokset vaikuttavat?
Millaisia vaikutuksia muutoksilla on eri sidosryhmiin?
Kuinka iso työ automaation lisääminen on?
Millaisia resursseja homman tekeminen vaatii?

Kuva 14. Lista ohjelmistoautomaation esisuunnittelu vaiheen kysymyksistä

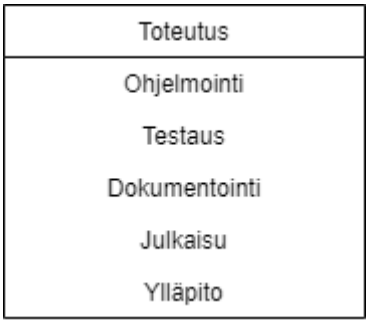
Kuvassa 14 on listattu kysymyksiä, joihin olisi hyvä olla edes jonkinlainen vastaus ohjelmistoautomaation esisuunnittelu-vaiheen jälkeen. Mitä tarkempi ja parempi vastaus, sitä helpompi on arvioida ohjelmistoautomaation toteuttamisen kannattavuutta.

Kannattavuuden arviointi
Hyödyt/Haitat
Tilanne nyt
Tilanne myöhemmin

Kuva 15. Kannattavuuden arvioinnin sisältämät vaiheet.

Yläpuolella kuvassa 15 on esitetty kannattavuudeen arvioinnin siältämiä vaiheita. Kun arvioidaan ohjelmistoautomaation kannattavuutta, perehdytään esisuunnitteluvaiheessa saatuihin tietoihin hyödyistä ja haitoista. Arvioidaan analyyttisesti, onko ohjelmistoautomaation toteuttaminen oikeasti järkevää ja kannattavaa. Ohjelmistoautomaation kannattavuuden arvioinnin päätyessä toteuttamisen kannalle, voidaan ohjelmistoautomaatio toteuttaa. Muutoksien julkaisemisen

jälkeen on järkevää kysellä käyttäjäpalautetta muutoksista. Täten muutoksien toteuttamisen järkevyys voidaan varmistaa vielä jälkeenpäin tai huomata missä vaiheessa päätöksentekoa mentiin väärään suuntaan. Tilanne myöhemmin -vaihetta voidaan käyttää myös sellaisissa tilanteissa, joissa toteutus todetaan juuri tällä hetkellä ei kannattavaksi, mutta tilanne voi muuttua tulevaisuudessa.



Kuva 16. Toteutuksen sisältämät vaiheet.

Kuvassa 16 on esitetty vaiheita, joita toteutus pitää sisällään. Toteutuksessa ohjelmoidaan muutokset suunnitelmien mukaisesti. Ohjelmoinnin jälkeen on muutokset järkevää testata ja varmistaa niiden toimivuus. Ohjelman dokumentaation päivittäminen muutoksien osalta on myös järkevää. Ohjelmoinnin ja testauksen ollessa valmiita voidaan ohjelman päivitykset julkaista käyttäjien käyttöön. Ohjelman ylläpidon vaatimukset voivat muuttua muutoksien takia.

Prosessin kehittämisellä tarkoitetaan prosessin parantamista entuudestaan. Prosessin kehittämisen suhteen on kannattavaa miettiä viimeistään prosessin jälkeen, voisiko prosessissa olla parannuskohteita. Toki parannuskohteita voi tulla esille myös prosessin aikana. Tullessa esille mahdollisia parannuskohteita kannattaa niille miettiä järkevä tapa ja paikka lisätä ne prosessiin. Täten prosessi kehittyy ja siitä tulee entistä parempi.

Osana prosessin kehittämistä voidaan pitää myös muun muassa, kuinka usein prosessia pitäisi toistaa tai kuinka pitkälle prosessissa toteutettava automaatio pitäisi viedä. Nämä ovat hankalia kysymyksiä, joihin ei ole yhtä oikeaa vastausta, joka olisi sopiva jokaiseen tilanteeseen. Tämä johtuu pitkälti jokaisen tapauksen olevan erilainen. Tosin prosessin toistamiseen voidaan todeta olevan tarvetta, mikäli ohjelmistoprosessi työllistää paljon tai ohjelmistoprosessin aikana tulee suhteellisen paljon mielikuvituksettomia ja tylsiä toimenpiteitä. Automaation suhteen, se kuinka pitkälle se kannattaa viedä, riippuu hyvin pitkälti siitä, mikä on järkevää siinä tietyssä tilanteessa. Mitä pidemmälle automaatio viedään, sitä vähemmän ihmisellä on kontrollia vaikuttaa asian lopputulokseen. Tämä on asia, joka kannattaa pitää mielessä miettiessä, kuinka pitkälle

automaatio kannattaa viedä, koska automaation mennessä pieleen voi siitä aiheutuvat seuraamukset olla sitä isommat, mitä vähemmän ihmisellä on alun perinkään ollut vaikutusta lopputulokseen.

6 Esimerkkiprojekti

Opinnäytetyön yhteydessä toteutettiin esimerkkiprojekti, jossa automatisoitiin ohjelmistoprosessi hyödyntäen kappaleessa 5 esitettyä prosessia. Projektissa automatisoitiin tiettyjen ohjelmistojen päivityspakettien siirtäminen oikeisiin kansioihin. Aikaisemmin tämä homma on täytynyt hoitaa käsin. Tästä johtuen päivityspaketteja on saattanut jäädä matkan varrelle tai mennä vääriin kansioihin. Tämän lisäksi käsin tehtynä hommaan menee turhaa aikaa, jota voitaisiin käyttää johonkin muuhun hyödylliseen.

Projekti alkoi kuvan 12 mukaisella tavalla eli nykyisen tilanteen arvioinnilla, jossa kartoitettiin, millaiselle automaatiolle olisi tarvetta tässä tietyssä tapauksessa. Ensimmäisenä asiaa lähdettiin tutkimaan mahdollisen kansioiden tarkastusohjelmiston kautta, joka olisi ilmoittanut päivityspakettien vääristä paikoista. Tosin nopeasti kävi ilmi, että päivityspakettien automaattinen siirtäminen olisi huomattavasti parempi ratkaisu.

Todettua päivityspakettien automaattisen siirtämisen asiaksi, jolle olisi eniten tarvetta tässä tapauksessa. Ruvettiin sitä esisuunnittelemaan prosessin mukaisesti. Esisuunnittelussa selvitettiin pääasiassa, miten automaatio toteutetaan ja minne automaatiota varten tehtävä koodi lisätään. Näiden lisäksi esisuunnitteluvaiheessa saatiin selville vastaukset kuvan 14 muihinkin kysymyksiin. Esisuunnittelu kokouksessa oli mukana sidosryhmistä ne henkilöt, joilla oli vaikutusta projektin lopputulokseen.

Ohjelmistoautomaation kannattavuuden arviointi oli tässä tapauksessa suhteellisen simppeli. Kuten aikaisemmin todettiin, ennen ohjelmistoautomaatiota ohjelmistojen päivityspakettien siirtäminen oikeisiin kansioihin on täytynyt hoitaa jokaisen itse manuaalisesti. Itse manuaalisesti hoidettaessa on ollut mahdollista tehdä virheitä, kuten siirtää paketteja vääriin kansioihin tai unohtaa tehdä koko pakettien siirto-operaatio. Virhetapauksien vähenemisen lisäksi ohjelmistoautomaation puolesta puhuu myös säästetty aika, joka on aikaisemmin mennyt homman manuaaliseen tekemiseen sekä tietysti virhetapauksien väheneminen säästää pitkällä tähtäimellä aikaa. Tässä tapauksessa ohjelmistoautomaation aiheuttamia haittoja ovat käytännössä vain ohjelmistoautomaation tekemiseen ja käyttöönottoon käytetyt ajat sekä resurssit. Muut mahdolliset haitat pystytään estämään tekemällä ohjelmistoautomaatio hyvin. Tälläisiä mahdollisia haittoja ovat muun muassa kansion tyhjennys -kutsun kutsuminen väärässä kansiossa tai päivityspaketin siirtäminen väärään kansioon. Kaikki asiat huomioon ottaen ohjelmistoautomaation toteuttaminen todettiin kannattavaksi ilman sen suurempia

vastaväitteitä. Todettua ohjelmistoautomaation toteuttaminen järkeväksi operaatioksi, prosessin mukaan jäljellä on itse ohjelmisto automaation toteuttaminen.

6.1 Projektin ohjelmistoautomaatio

Ohjelmistoautomaatio toteutettiin liittämällä päivityspakettien siirtäminen Jenkin-automaatiopalvelimeen. Jenkins on itsenäinen, avoimen lähdekoodin automaatiopalvelin, jota voidaan käyttää kaikenlaisten ohjelmistojen rakentamiseen, testaamiseen ja toimittamiseen tai käyttöönottoon liittyvien tehtävien automatisoimiseen [16]. Tässä tapauksessa Jenkins tekee päivittäin uuden kansiopaketin ohjelmistosta ja Jenkinssissä pystytään katsomaan, onko uudessa paketissa tullut päivityksiä vai ei. Kun päivityksiä on tullut, kutsutaan tätä automaatiota varten tehtyä koodia ja Jenkinssin kautta annetaan koodille parametrina, mitä koodissa tarvitsee ladata sekä mihin julkaisukansioon tarvitsee tarvittavat zip-tiedostot siirtää. Koodissa ensin ladataan kansiopaketti ja haetaan kansiopaketin sisältä kaikki zip-tiedostot, jotka täytyy kopioida julkaisu kansion omiin sisäkansioihinsa. Mikäli julkaisu kansiossa ei ole tarvittavia sisäkansioita, tekee koodi ne ennen zip-tiedostojen kopioimista. Ennen joidenkin zip-tiedostojen kopioimista, pitää osa julkaisu kansion sisäkansioista tyhjentää, jotta ne sisältävät operaation jälkeen vain uusimman tiedoston, joka operaatiossa siirrettiin sinne. Lopuksi tehdään kansiopaketista oma zip-tiedosto, joka kopioidaan omaan sisäkansioonsa julkaisukansiossa ja poistetaan aluksi ladattu kansiopaketti sekä tehty zip-tiedosto.

Isoimmat sudenkuopat tämän automaation tekemisessä oli varmistaa asioiden tapahtuminen oikeissa kansioissa, kuten esimerkiksi julkaisukansion sisäkansion tyhjentämis- kutsun kutsuminen väärässä kansiossa voisi pahimmassa mahdollisessa tapauksessa aiheuttaa monien kansioiden ja tiedostojen poistamista, mikä ei ole tietenkään automaation suunniteltu ominaisuus. Tämän lisäksi automaation siirtäessä zip-paketit vääriin kansioihin voisi aiheuttaa hämmennystä ja turhaa manuaalista työtä, jossa tarvitsisi käydä manuaalisesti kansioita läpi ja siirrellä vääriin kansioihin menneet asiat oikeisiin kansioihin.

6.2 Automaation arvo

Yleisesti ottaen automaation arvon arvioimista on käyty läpi jo aiemmin kappaleessa 3.3 sekä esimerkkiprojektin automaation arvoon vaikuttavia tekijöitä on hieman sivuttu jo kappaleissa 6

ja 6.1. Tämän projektin osalta automaation arvoon vaikuttavat tekijät ovat virheiden väheneminen, automaation kautta vapautuva aika ja ainakin jossain määrin tylsän työtehtävän poistuminen työlistalta.

Kuten parissa aiemmassa kappaleessa aihetta sivuttiin, niin automaation kautta virheiden tapahtuminen tämän ohjelmistoprosessin osalta vähenee. Aiemmin ohjelmistoprosessia manuaalisesti tehtäessä on voinut sattua inhimillisiä virheitä zip-paketteja kopioitaessa kansioihin. Nämä virheet jäävät entistäkin vähemmiksi tai jopa kokonaan pois, kun automaatio on toteutettu hyvin ja suunnitelmien mukaisesti.

Vapautuvan ajan täysin todellista arvoa on erittäin hankala arvioida, kuten kappaleessa 3.3 todettiin, mutta nyt tämän projektin kautta automatisoituun ohjelmistoprosessiin aiemmin käytetyt minuutit ja sekunnit vapautuvat tulevaisuudessa muuhun käyttöön. Mahdollisesti ne voidaan käyttää entistäkin luovempaan työhön, mikä olisi kaiken kaikkiaan hyvä asia, kuten kappaleessa 3.1 havaittiin.

Kappaleessa 3.1 todettiin, että annetaan koneiden tehdä tylsät, mielikuvituksettomat ja toistuvat työtehtävät, koska koneet pystyvät hoitamaan ne ihmistä paremmin sekä ne kuluttavat ihmistä loppuun enemmän. Nyt tässä esimerkkiprojektissa automatisoitu ohjelmistoprosessi kuuluu ainakin jossakin määrin tähän kategoriaan. Tästä johuten tämän ohjelmistoprosessin poistuminen manuaalisesti toteutettavien ohjelmistoprosessien listalta on hyvä asia.

Kaiken kaikkiaan esimerkkiprojektissa automatisoitu ohjelmistoprosessi tuo paljon positiivisia asioita, aiheuttamatta isompia ongelmia, mikäli kappaleessa 6.1 mainitut sudenkuopat on vältetty. Ainoa hieman isompi asia, mikä automaatiosta aiheutuu, on automaation päivittäminen, jos julkaisukansion kansiorakenne muuttuu tulevaisuudessa. Toisaalta julkaisu kansion kansiorakenteen muuttuminen aiheuttaisi lisää muistettavia asioita ohjelmistoprosessia manuaalisesti tehdessä, jonka takia inhimilliset virheet voisivat ainakin hetkellisesti lisääntyä. Tämän takia ohjelmistoprosessin automatisointi ei välttämättä ollut huono asia myöskään tältä osin, koska automaation päivittämisen jälkeen asiat pitäisivät taas toimia kuten ennen julkaisukansion kansiorakenteen muuttumista. Totta kai automaation päivittäminen aiheuttaa hetkellisesti lisää töitä, mutta päivityksen jälkeen automaatio tuo samat hyödyt kuin aikaisemmin.

7 Prosessin arviointi

Prosessin arvioinnissa arvioitiin kuvan 12 prosessia, joka esitettiin kappaleessa 5. Prosessia myös käytettiin kappaleen 6 esimerkkiprojektissa. Arvioinnissa tutkittiin prosessin yleistä toimivuutta ja millä tavoin prosessia voisi mahdollisesti parantaa entistäkin paremmaksi.

Esimerkkiprojektin perusteella voidaan todeta prosessin toimivan pitkälti, miten sen on suunniteltu toimivan. Nykyisen tilanteen arviointi -vaiheessa tarkoituksenahan on kartoittaa, mille ohjelmistoautomaatioille olisi oikeasti tarvetta. Esimerkkiprojektissa oli muutama idea millaiselle ohjelmistoautomaatiolle voisi olla tarvetta, joista lopulta toteutettu automaatio todettiin kaikista järkevimmäksi.

Esisuunnitteluvaiheessa saatiin selville vastauksia kysymyksiin, joita sitä ennen oli. Tämä helpotti huomattavasti itse automaation toteutusta sekä esisuunnitteluvaiheen kokouksessa mukana olleet henkilöt olivat tietoisia tälläisen automaation lisäämisestä lähitulevaisuudessa.

Ohjelmistoautomaation kannattavuuden arviointi hoiti kyllä osansa prosessissa esimerkkiprojektissa, vaikka tässä tapauksessa ei ihan hirveästi tarvinnut ruveta arpomaan kannattavuuden osalta. Sellaisissa tapauksissa, jossa automaatiosta aiheutuu myös negatiivisia asioita, joudutaan pohtimaan kannattavuutta paljon enemmän. Itsessään automaation arviointia on aikaisemmin avattu kappaleessa 3.3.

Prosessin viimeisessä vaiheessa eli automaation toteuttamisessa ei ollut mitään sen ihmeellisempää, koska mikäli prosessia on noudatettu, pitäisi automaation toteuttamiseen olla suhteellisen selvät sävelet esisuunnitteluvaiheessa saatujen tutkimusten perusteella. Tämän takia teorian muuttaminen käytäntöön pitäisi olla huomattavasti helpompaa verrattuna tilanteeseen, jossa teoriaa lähdettäisiin toteuttamaan ilman sen suurempia suunnitteluja tai tutkimuksia aiheesta.

Automaation tuomien muutoksien julkaisemisen jälkeen kannattaa kysellä käyttäjiltä palautetta tehdyistä muutoksista. Täten voidaan varmistua toteuttamisen olleen oikea päätös tai saadaan tietoon, minkä takia muutoksien toteuttaminen ei ollutkaan järkevää. Tällä tavalla kannattavuuden arviointia voidaan kehittää prosessissa paremmaksi, kun huomataan, missä vaiheessa päätöksen teossa mentiin väärään suuntaan.

Yleisesti ottaen ainakin esimerkkiprojektin osalta prosessi tuntui toimivan varsin hyvin. Mahdollisesti kunnon sidosryhmien analyysi, jota käytiin läpi kappaleessa 4.2 voisi olla tarpeellinen toimenpide, jota pitäisi korostaa entistäkin enemmän prosessissa. Tosin tämän toimenpiteen voi suorittaa entuudestaan esimerkiksi nykyisen tilanteen arvioinnissa, kun selvitetään projektiin liittyvät sidosryhmät. Useimmissa tapauksissa sidosryhmien analysointi todellakin on kannattava operaatio, josta projektin toteuttamisen aikana on hyötyä. Sidosryhmien analysointia voidaan korostaa esimerkiksi lisäämällä nykyisen tilanteen arviointiin oman maininnan siitä. Täten sidosryhmien analysointi olisi osa nykyisen tilanteen arviointiin kuuluvia toimenpiteitä. Tämän lisäksi prosessin arviointia kannattaa toteuttaa aina prosessin jälkeen.

8 Yhteenveto

Ohjelmistoprosessi on joukko toisiinsa liittyviä toimintoja, jotka johtavat ohjelmiston tuotantoon. Tämä perustuu päätösten tekemiseen. Ohjelmistoprosessien kehittämisellä pyritään toteuttamaan parannustoimia tiettyjen tavoitteiden saavuttamiseksi, kuten tuotantonopeuden lisääminen, korkeamman tuotelaadun saavuttaminen tai kustannusten alentaminen.

Ohjelmistoprosessin automatisoinnilla on tapauskohtaisesti erilaisia positiivisia vaikutuksia, mutta toisaalta automatisoinnista voi aiheutua negatiivisia asioita. Täten automaation lisääminen ohjelmistoprosessiin kannattaa olla harkittu päätös eikä hetken mielijohteesta aiheutuva teko. Ennen päätöstä automaation toteutuksesta on hyvä tehdä taustatutkimusta aiheeseen liittyen. Tällä tavalla päätöksen tekeminen automaation kannattavuudesta on huomattavasti helpompaa sekä päätöksen päätyessä kannattavuuden puolelle on teorian toteuttaminen paljon vaivattomampaa.

Automaation toteuttamisessa on tärkeää ottaa huomioon sidosryhmät. Sidosryhmien huomiotta jättäminen aiheuttaa hyvin todennäköisesti ongelmia tulevaisuudessa. Tästä aiheutuvat ongelmat voivat johtaa lisäkuluihin tai pahimmassa tapauksessa kokonaan hukkaan menneeseen sijoitukseen.

Tässä työssä kehitetty prosessi automaation lisäämisestä ohjelmistoprosessiin sisältää neljä syklistä vaihetta. Nykyisen tilanteen arvioinnissa tutkitaan ohjelman nykyinen tilanne ja mahdollisuudet automaatioille. Ohjelmistoautomaation esisuunnittelussa perehdytään tarkemmin mahdollisiin automaatioihin. Kannattavuuden arvioinnissa päätetään automaation toteuttamisesta. Kannattavuuden arvioinnin päätyessä toteuttamisen kannalle voidaan automaatio toteuttaa. Luotua prosessia käytettiin esimerkkiprojektissa, jossa prosessin toimivuus todettiin varsin toimivaksi.

Lähteet

1 Elgabry, Omar. (2017). Software Engineering – Software Process and Software Models (Part 2). Saatavilla osoitteesta: https://medium.com/omarelgabrys-blog/software-engineering-software-process-and-software-process-models-part-2-4a9d06213fdc. Haettu 2.12.2021

2 Zobrist, Steve. (2014). The Importance of Software Process Improvement – It's the journey, not just the destination. Saatavilla osoitteesta: https://blogs.perficient.com/2014/10/10/software-process-improvement/. Haettu 22.11.2021

3 Sami, Mohamed. (2018). The Software Process Improvement (SPI) – Reward or Risk. Saatavilla osoitteesta: https://melsatar.blog/2018/06/26/the-software-process-improvement-spi-reward-or-risk/. Haettu 3.12.2021

4 What is Prcoess Automation? Saatavilla osoitteesta: https://www.tibco.com/reference-center/what-is-process-automation. Haettu 22.11.2021

5 Fogg, Erik. (2019). Software Development Automation: What Should We Automate? Saatavilla osoitteesta: https://prodperfect.com/blog/test-development/automated-software-development/. Haettu 24.11.2021

6 Kaushik, Lata. (2020). Advantages And Disadvantages Of Automation. Saatavilla osoitteesta: https://mylifemyfiction.com/automation/advantages-and-disadvantages-of-automation/. Haettu 3.1.2022

7 gpadvisors, (2017). Process Automation: Advantages and Disadvantages for IT Companies. Saatavilla osoitteesta: https://www.gb-advisors.com/process-automation/. Haettu 31.12.2021

8 Babb, Benjamin. (2021). How to Automate Any Process. Saatavilla osoitteesta: https://www.pipefy.com/blog/automate-manual-process/. Haettu 25.11.2021

9 salesforce, (2020). IT Leaders Fueling Productivity With Process Automation. Saatavilla osoitteesta: https://www.salesforce.com/content/dam/web/en_us/www/documents/platform/it-leaders-fueling-time-and-cost-savings-with-process-automation.pdf. Haettu 5.1.2022

10 concepta, (2018). How To Define Stakeholders For your Software Development Project. Saatavilla osoitteesta: https://www.conceptatech.com/blog/how-to-define-stakeholders-for-your-software-development-project. Haettu 24.11.2021

11 Minning, Lauren. (2021). The 10 Types of Stakeholder That You Meet in Business. Saatavilla osoitteesta: https://www.activecampaign.com/blog/types-of-stakeholders. Haettu 17.1.2022

12 Stakeholders. Saatavilla osoitteesta: https://osome.com/uk/term/stakeholders-uk/. Haettu 17.1.2022

13 ProjductPlan. Stakeholder Analysis. Saatavilla osoitteesta: https://www.productplan.com/glossary/stakeholder-analysis/. Haettu 18.1.2022

14 The Project Management Blueprint.com, (2019). Analyzing and Classifying Project Stakeholders. Saatavilla osoitteesta: https://www.theprojectmanagementblueprint.com/blog/stakeholder-management/stakeholder-power-interest-grid. Haettu 18.1.2022

15 Jo Can Do, (2020). The power-interest grid. Saatavilla osoitteesta: https://jocando.co.uk/power-interest-grid/. Haettu 1.2.2022

16 Jenkins User Documentation. Saatavilla osoitteesta: https://www.jenkins.io/doc/. Haeuttu 9.2.2021