HOW TO CHOOSE BETWEEN ROBOTIC PROCESS AUTOMATION AND BACK-END SYSTEM AUTOMATION?

Research paper

Penttinen, Esko, Aalto University, Helsinki, Finland, esko.penttinen@aalto.fi

Kasslin, Henje, Aalto University, Helsinki, Finland, henje.kasslin@aalto.fi

Asatiani, Aleksandre, Aston University, Birmingham, UK, aleksandre.asatiani@iki.fi

Abstract

Recent literature has identified two main types of IT-enabled innovation and labelled them as lightweight IT and heavyweight IT. In this study, we look into robotic process automation (RPA) as lightweight IT and traditional back-end system automation as heavyweight IT and study how a case company Telco makes the choice between these two alternative ways of implementing automation. Drawing on an empirical study conducted at Telco examining two automation projects, we find support for earlier studies addressing specifically the choice problem of RPA. Our main contribution lies in uncovering the role of the two-dimensional feature stability on the choice. While heavyweight IT is preferred when system architecture is stable, lightweight IT, operating on the presentation layer, holds a prerequisite of stable system interfaces.

Keywords: Robotic process automation, business process, lightweight IT, heavyweight IT, system architecture.

1 Introduction

To stay competitive, companies need to constantly search for ways to improve their business processes. Despite the long academic and practitioner-oriented tradition of business process development, there still exist plenty of ill-handled processes in firms. Indeed, the market for business process development is expected to further grow from an estimated USD 6.96 bn in 2016 to USD 13.52 bn in 2021[1]. Once a task susceptible for improvement has been identified, information technology (IT) offers several alternative paths for improving the task through business process automation. Recent literature identifies two approaches to process automation: through lightweight IT and through heavyweight IT (Bygstad, 2016). Heavyweight IT refers to a knowledge regime driven by IT professionals, enabled by proven digital technology, and realized through software engineering. Lightweight IT, on the other hand, is a socio-technical knowledge regime, driven by competent users' need for solutions, enabled by the consumerization of digital technology, and realized through innovation processes (Bygstad, 2016). Typical examples of these alternative regimes are traditional back-end system automation (integrated through Application Programming Interface (API)) for heavyweight IT and graphical user interface (GUI) automation (such as robotic process automation (RPA)) for lightweight IT (Bygstad, 2016; Lacity and Willcocks, 2016). The lightweight IT approach tends to be faster and cheaper to implement, however, before making the decision to implement

[1] Business Process Management Market worth 13.52 Billion USD by 2021
http://www.marketsandmarkets.com/PressReleases/business-process-management.asp

lightweight IT, one needs to consider issues such as suitability of the business process to GUI automation, technological limitations of GUI automation, and long-term impact of the approach on business (Lacity *et al.*, 2016; Willcocks *et al.*, 2015a). Therefore, we call for more nuanced discussion on the decision problem of choosing between lightweight and heavyweight IT. This study seeks to answer the following question: *How does a company choose between the implementation of lightweight IT and heavyweight IT?*

We use RPA as an example of lightweight IT in this study. RPA is a relatively new breed of automation software configured to emulate a human worker by interacting with information systems through existing user interfaces (Institute for Robotic Process Automation, 2015; Lacity and Willcocks, 2016). Building on the recent literature on lightweight and heavyweight IT and drawing on a single case study where both approaches were considered, we find support for earlier studies' findings on selection criteria for RPA. Considering the choice between back-end system automation and RPA, our main finding concerns the stability of the environment where existing systems operate. In this case, stability refers to the degree of change within the components of an IT system (e.g. IT architecture and user interfaces). For example, a stable user interface would mean that there are very few changes to the design or layout of the user interface during the system's lifecycle. Based on our analysis, we found that while heavyweight IT requires stable system architecture, it is not sensitive to user interface changes on the presentation layer. Inversely, lightweight IT requires stable user interfaces, but is relatively insensitive to system architecture changes.

2 Lightweight IT vs heavyweight IT

The comparison of lightweight and heavyweight IT was developed by Bygstad (2016), who introduced these terms for dealing with two trends in IT industry. The first trend he identifies is the growing size and interconnectivity of IT systems, partly driven by the attempt to integrate separate IT systems. With integrations, companies can try to reduce complexity, lack of agility, and hindrance for innovation that are caused by siloed systems. The second trend is consumerization, a development that challenges the hegemony of IT departments with trends such as 'bring your own device', where technology bypasses the IT departments. This, Bygstad believes, is a response to bureaucratic solutions and mechanisms of company-wide IT.

Lightweight and heavyweight IT are not viewed merely as different technological approaches, but two different knowledge regimes. A knowledge regime in this context means a unity that includes a network of actors, work practices, technologies and the shared knowledge. Heavyweight IT presents more or less the "traditional" or "mainstream" IT delivery, whereas lightweight IT is characterized by business orientation, quick experimentation, and user driven solutions bypassing IT departments and utilizing easily available technologies (Bygstad, 2016). Another key difference in lightweight and heavyweight IT is the degree of their invasiveness. Lightweight solutions often use the presentation layer and do not change the underlying deep structure of the system or data architectures, whereas heavyweight solutions act on the data access or business logic layers (Willcocks *et al.*, 2015a).

Heavyweight IT is defined as a knowledge regime that is associated to the development and maintenance of large systems (Bygstad, 2016). The typical profile of heavyweight IT systems is back-end and transactional, often supporting the documentation of work. In heavyweight IT, more sophisticated solutions are developed through sometimes complex integration. Therefore, it is the core territory of IT professionals and proven technologies; back-end integration typically requires software engineering and the utilization of solutions such as enterprise resource planning systems (ERP), service-oriented architecture (SOA), or an enterprise service bus. Heavyweight IT is a mature field, and it has a systematic development culture that concentrates heavily on quality and security.

Despite its established nature, heavyweight IT has its own challenges. As Sommerville et al. (2012) point out, the traits of growing scale and integration that are characteristic for heavyweight IT often lead to increasing costs and complexity. Moreover, in many organizations, heavyweight IT is facing more requests for development than it can fulfil, which leads to growing IT development backlogs (Bygstad, 2016; Willcocks *et al.*, 2015a).

In contrast, lightweight IT is defined as the *new* knowledge regime that is closely related to the consumerization of IT. The profile of lightweight IT is typically applications that support business processes without modifying the original IT architecture. As a result, its essential feature is that its deployment often bypasses IT departments. The development of lightweight IT relies more on competent, non-IT users and vendors, who can create simple, specialized applications to support work tasks. Cheaper, more available and easy to use technology is at the core of lightweight IT: lower cost and accessible technologies offer non-specialists new experimenting possibilities. In a sense, lightweight IT can act as an “innovation arena” (Bygstad, 2016). However, also lightweight IT has its problems: lightweight solutions can lead to disconnected applications and gadgets, and also privacy and security issues can be harder to address with lightweight technologies. We refer to Table 1 for a comparison between lightweight and heavyweight IT.

Feature	Lightweight IT	Heavyweight IT
Type of systems	GUI automation	Back-end systems automation
Technology	Emergent, spontaneously adopted	Mature, proven
Culture	Business and process improvement	Software engineering
Focus	Agility, innovation, speed	Security, efficiency, reliability
Application area	Unknown, development of new services	Well-understood and known services
Invasiveness	Non-invasive, presentation layer	Invasive, data-access and business logic layer
Problems	Isolated systems, privacy and security issues	High complexity and costs of systems

Table 1. Comparison of lightweight and heavyweight ITs

2.1 Lightweight and heavyweight IT in practice

We study automation of IT-enabled business processes, concentrating on two different approaches: RPA and back-end system automation. We define back-end system automation as invasive automation, implemented by means of system development and/or data or application layer system integration. Automation typically requires either system development or integration of disconnected systems, or both. Information systems are typically divided into three layers: data layer, application layer, and presentation layer (Forrester, 2011; Manuel and AlGhamdi, 2003) The main difference of presentation layer integration compared to the others is that it reuses existing system functionalities, whereas the first two require changing the system logic or interfaces. Presentation layer integration is independent of the underlying architectural approach: no matter whether the system is developed with Java or COBOL, user interface already enables accessing the underlying data (Lacity and Willcocks, 2016). While data and application layer integration are more scalable and efficient, they require higher technology sophistication (Forrester, 2011). This is the case with back-end system automation: it requires specialized knowledge and skills on the field of IT and is invasive, thus being an example of heavyweight IT. In contrast, RPA operates mainly on the presentation layer and already existing functionalities of applications and, therefore, can be viewed as prime example of lightweight IT (Institute for Robotic Process Automation, 2015; Lacity and Willcocks, 2016).

2.1.1 Robotic Process Automation as lightweight IT

Robotic Process Automation is an emerging field of business process automation. It differs profoundly from back-end system automation approaches in the aspect that it utilizes the existing features and user interfaces of already existing systems (Lacity and Willcocks, 2016). When implementing RPA, no underlying systems programming logic needs to be changed as systems are accessed through the presentation layer (Willcocks *et al.*, 2015a). In RPA, applications called 'software robots' are configured to interact with information systems, replicating the actions of human workers (Hallikainen *et al.*, 2018; Lacity and Willcocks, 2016).

In RPA terminology, a "robot" is an equivalent to a single software instance (Lacity and Willcocks, 2016). Different tools and practices exist for configuring robots to perform the desired processes: these include recording workflows, using graphical interfaces with process flowcharts or using scripting language (Willcocks *et al.*, 2015a). Process libraries can be created out of the modelled process elements which contain detailed step-by-step instructions that robots follow (Asatiani and Penttinen, 2016). This way, elements used in multiple processes, like logging in to a certain system, can be reused and updated more easily.

RPA has various benefits. First, it is a relatively cost-effective method of process automation. With RPA, automation can be implemented in a short timeframe (usually from couple of weeks to maximum of few months), allowing for a fast return on investment. RPA enables automation of processes, which have been previously considered too costly to automate (Barnett, 2015). Typically, the costs of a software robot are between 10%-19% of an in-house full-time equivalent (FTE) and between 33%-50% of an offshore FTE (Prangnell and Wright, 2015; Slaby, 2012; Willcocks *et al.*, 2015a). Moreover, it has been claimed that RPA software with its graphical interface is designed for non-programmers and advanced IT skills are not needed in robotizing processes (Willcocks *et al.*, 2015a). This would enable business users to create quick solutions themselves and allow for agility in changing business processes (Forrester, 2011). Examples of situations where this type of quick 'pontoon bridge'-solution could be useful include sudden changes in business environment, or business users wishing to make simple extensions to existing applications (Forrester, 2011). RPA could also be utilized in testing new business strategies cheaply and quickly (Willcocks *et al.*, 2015a). Finally, as RPA is non-invasive by nature, it enables connecting systems where interfaces are hard or impossible to build: for example, creating interfaces to legacy applications, where a costly and difficult software reengineering would otherwise be needed. These interfaces can, in turn, be used, for example, in business process management (BPM) solutions (Barnett, 2015).

However, RPA has its limitations. As RPA is a software-based solution, the inputs must be digital and the process must be rule-based. Software robots can, for example, move and click the mouse, interpret text and pictures on screen, and copy, paste and write characters. RPA is technology independent, and robots can use any applications and sources from the mainframe to Excel and from CRM or ERP to web applications (Willcocks *et al.*, 2015a). This, however, assumes existing infrastructure built on heavyweight IT tools. Other lightweight solutions, like macros and scripting, have existed already for decades. However, RPA can be seen as an evolution from these basic tools, as enterprise RPA platforms allow defining a lot richer logic and support more complex processes. Compared to these solutions, the most notable potential of RPA lies in automating processes that are deeper in the traditional domain of knowledge workers. Nevertheless, it is important to emphasize that RPA tools are unable to make decisions or adapt to the changing environment. Therefore, RPA is most valuable in strictly defined, low cognitive, high-volume tasks (Aguirre and Rodriguez, 2017). Although software robots outperform humans in terms of speed and quality and make no errors, they cannot work faster than the overall process allows them to. Although robots could theoretically be working 24/7, the existing processes can considerably limit the effective time (Willcocks *et al.*, 2015a). Also, although RPA is claimed to be business driven, IT still has an important role in setting up the facilities: this includes access rights, maintaining process logs, and setting up virtual resource agents. Encryption might be needed when robots use the existing network infrastructure to access the existing applications (Forrester, 2011). Maybe most importantly, RPA is inferior to back-end system automation when it comes to performance: RPA is not optimal for handling the heaviest transaction masses, and, as a result, currently it is mostly a temporary solution bridging the gap between large-scale system development and running manual processes on legacy systems (Asatiani and Penttinen, 2016).

2.1.2 Back-end system automation as heavyweight IT

According to Mohapatra (2009), there are four basic ways of implementing business process automation. These are 1) extending the current system, 2) purchasing a BPM-solution with business process automation (BPA) extension, 3) purchasing a middleware solution, or 4) using special purpose built-in tool. At least the first three fall to the category of back-end system automation. By extending the current system, companies can sometimes sufficiently respond to the new business needs. This means extending the features of already existing system, typically by means of system development. Extending the current system can be sufficient when automation is done in process using one system. Purchasing Business Process Management solution and developing BPA extension is feasible, although the two are distinctively different. However, this approach can be time consuming, as although BPM provides a framework for mapping all business processes, this delays automation of individual processes. The third alternative is purchasing a middleware solution. The term middleware is used to describe solutions that are used to connect two or more already existing programs. They are a central element of IT infrastructures, as they allow joining heterogeneous systems together in one framework. This is done by routing data between databases and end user applications (Mohapatra, 2009). Multiple middleware approaches have been developed, one of the newer ones being Enterprise Application Integration (EAI), which provides an integration framework for combining a set of technologies (Sabooniha *et al.*, 2012). Serrano et al. (2014) offer a comparison of solutions for EAI. They group these solutions from lightest to heaviest as follows: Integration framework (libraries implementing APIs), Enterprise Service Bus (ESB, frame and deployment, administration and monitoring tools), and Integration suite (ESB and business specific tools like BPM). The most suitable approach depends on complexity and requirements. The final approach is acquiring special purpose built-in tool. Mohapatra (2009) sees these as shortcuts to automation, as they are built specifically for business process automation, and can often be used also by non-technical users due to simple user interfaces. However, they also add a new software provider into the equation.

2.1.3 Choice between RPA and back-end system automation

While the existing literature offers some insights on the choice criteria for selecting specifically robotic process automation, articulating features such as high volume of transactions, need to access multiple systems, stable environment, and limited need for exception handling (Fung, 2014; Willcocks *et al.*, 2015b), we note a lack of studies addressing the selection problem between RPA and alternative automation technologies. The problem with most of the criteria cited above is that they are independent of the type of approach to automation and they can be used to argue for both back-end system automation and RPA.

3 Methodology

We chose to approach the research question with a qualitative study using an inductive stance (Yin, 2015). We operationalize lightweight IT in terms of RPA, which is a relatively new field of automation. Thus, there is little academic research on the topic and an inductive stance can help to lay the groundwork for future research. Also, as the technology is still at the early adoption stage, getting sufficient quantitative data on decision-making criteria was unlikely.

We analysed two cases of RPA implementation within a large telecommunications company headquartered in Finland, named Telco (pseudonym). In both cases, RPA and back-end system automation were considered as alternatives. Interviews were an important source of evidence, complemented by document analysis and participant observations made during the data collection period. For both cases, we documented and analysed the process before and after implementing RPA and discussed the factors that affected the selection of RPA. This setup can be viewed as multiple case design. In this case, the purpose was to explore the range of factors affecting the company's decisions. The selected cases were different by nature and provided different insights.

The data was collected over a period of three months during spring 2017. Data collection included two main phases, which are described in Table 2. The first phase included unofficial discussions and interviews with Telco during which we identified potential informants and RPA projects for case studies. The findings were recorded in notes and e-mail discussions, and they served in planning the main data collection round.

The second phase consisted of data collection for Telco case studies, where interviews, documentation and participant observations were combined to generate in-depth understanding about the processes and decisions. All interviews in Phase II were recorded and transcribed for analysis.

Phase	Method and Informant	Outcome
Phase I. Identifying cases and informants	Informal discussion with Head of department	Identified cases and informants
	Informal discussion with Solution area architect	Identified case 2
	Informal discussion with Various attendants	Identified informants
	Informal discussion with Various attendants	Info on status of pilots
	Interview with Service manager	Comparison of RPA and back-end
Phase II. Case interviews and case material collection	Interview with Service specialist, carrier services	Info on case 1
	Interview with Service manager, order handling	Info on case 2
	Interview with Service specialist, carrier services	Narrative of case 1
	Interview with Service manager, order handling	Narrative of case 2
	Review of Documents, access to RPA project documentation, process documentation and other material	In-depth info on RPA cases
	Participant observation, participating to meetings, informal discussions, e-mail	In-depth info on RPA cases

Table 2. Data collection

We iteratively analysed the transcribed data. The analysis was done in three main stages: creating an initial coding scheme (Stage 1), applying and developing the coding scheme (Stage 2), and analysis of the coded material (Stage 3). In the first stage, we went through the whole transcribed material and identified recurring themes from each interview. Having identified these initial themes, we combined them into an initial coding scheme (see Table A1 in the Appendix) that could be applied to the qualitative data corpus. In the second stage, the transcribed material was imported to Atlas.ti, where we coded the material using the scheme created during the first stage. More codes came up during the process, while some existing codes were combined and modified. After the first round of coding, we started iterating the process, looking for connections between codes: for example, if two codes appeared together 70% of the time, conclusion was that they should be combined or were tightly linked on a conceptual level. The coding scheme is available in Appendix. As a result, in the third stage, we could analyse the coded material and identify decision-making criteria and factors affecting the company's selection between RPA and back-end system automation.

4 Findings

In the two cases, a manual process was identified to which RPA and back-end system automation were considered as alternative approaches. RPA pilots were conducted during year 2016. RPA-enabled processes were taken into production during Spring 2017. Below, we use pseudonyms to refer to systems (italic) and companies involved in the cases.

4.1 Case 1: availability check of fibre-Ethernet product

The first automation case was related to a process in Telco's Corporate Customers-unit. The process was the availability check of the fibre-Ethernet product – in short, checking whether there is available capacity to sell when a bid for the Ethernet-product arrives. The availability check is only one phase of bid processing; most of the process was left for manual handling, as it requires expert insight.

The process had various aspects that made it a tempting target for automation. First of all, the fibre-Ethernet is one of the most frequently requested products in the unit, thus the process volumes are high, implying considerable potential benefits from automation. At the same time, the product and process are relatively simple: after the steps are mapped, conducting a capacity check is rather straightforward. Moreover, although human users utilize up to four different systems in the process, it was possible to complete it with two, which would make the implementation of any kind of automation easier.

One important gain expected from automation was the easing of the service specialists' work. Many checks need to be conducted during the process, and the goal was to find out whether automation could assist the experts.

4.1.1 Process before and after automation

Before embarking on the automation journey, the process was as follows: A customer bid arrives through a web portal to the order and bid delivery system *OrderSys*. The bid is automatically directed to the correct work queue according to the product, and a service specialist takes it for processing. The specialist checks basic information, including what was ordered and to which address. Next, he uses the address to find out which distribution point is serving it (in bigger locations there might be multiple distribution points). For this, specialists can use either the address register system *Adresso* or the map application *Maps*. From *Maps*, a specialist can visually see the building and relevant information: what type of distribution points there are, is there fibre or copper, is there a switch, and so forth. Also, after identifying the correct distribution point, she/he can see its ID. After finding the ID, the specialist opens the network information system *NetInfo* and finds the same distribution point from there (specialists typically do this using the ID, but it is also possible to find the distribution point using the address from *OrderSys*). Now, the specialist can see whether there is capacity available. The required information includes: is there fibre going to the location, is there capacity on the fibre, is there a switch at the location, and if so, are there available ports at the switch. After checking this information, the specialist is ready with the capacity check and can move to next phases of processing the bid.

The process above describes how a human user typically conducts the capacity check. However, the process can be completed using only *OrderSys* and *NetInfo*. This is because getting the distribution point ID from *Maps* or *Adresso* is not necessary: distribution point can be found from *NetInfo* using the address retrieved from *OrderSys*. However, for human users, *Adresso* and *Maps* offer more versatile visual information that makes processing the bid easier. For any kind of automation tool, this visual information does not add value.

A decision was taken to implement RPA in this process. After the implementation of RPA, the process was as follows: the software robot logs into *OrderSys*, opens the correct work queue and sorts the tasks by date from oldest to newest. Next, the robot opens the oldest ticket and checks the product from the description field. If the product is wrong, the robot moves to the next ticket - if the product is correct, the robot continues by checking the speed and address information. After this, the robot opens *NetInfo* and finds the correct distribution point using the address fetched from *OrderSys*. This requires feeding in the address, and then navigating the tree-structure of the system step by step: information in *NetInfo* is organized as a tree, where a click of a mouse opens branches. The tree structure proved challenging, as there can be a great number of branches: for example, if the fibre goes through multiple distribution points, the path can become very long. However, in the end it was possible to train the robot to navigate through the tree, finding and following the correct fibre trail to the end.

At the end of the trail, the software robot checks if there is free fibre capacity at the location. After this, the robot returns an answer in text format to *OrderSys*. It is important that the robot returns an informative answer for the human specialist, as this is needed for pricing the bid: is there fibre, is there a switch, are there ports and do these have free capacity. At its simplest, the answer can be "no fibre trail". Finally, after returning the information text, the robot moves the ticket to the next queue and continues with the next task in line.

No exceptions or errors were faced where the robot would return the task to manual handling. However, when the RPA is taken into production, it is likely that "no fibre trail" –answers would be double checked by human specialists to make sure the robot has checked all possible distribution points. Thus, big part of the process remains manual. Processing the bids requires expert judgement: after receiving availability check results, an expert accepts or rejects the bid, determines the pricing for accepted bids and finally answers to the customer. The robot's task is simply to provide background information for experts' decision making.

4.1.2 Decision

Prior to the RPA implementation for this process, alternative automation approaches were also explored. In this case, deciding between RPA and back-end system automation was not entirely straightforward. The considered back-end system automation approach was to be implemented through *AutoOp.*

Many Telco experts looked into the *AutoOp* -alternative, as it seemed very promising at first sight. *AutoOp* was already at use in other processes, and, therefore, it was tempting to utilize it also here: the support processes and organization were existing and the software was proven to work. Also, based on the initial investigations, it seemed likely that automating the process with *AutoOp* would be possible.

However, during the investigations, problems started to occur. The first challenge was a lack of interfaces: there was no existing interface between *OrderSys* and *NetInfo* so back-end system automation would have required development efforts in both systems, as well as building the interface. Building the interfaces would alone have taken several months and required lot of investments and the overall schedule did not seem feasible. Also, *AutoOp* was not able to use *NetInfo*, but only the databases behind it: not all data could have been fetched from the databases without extra work. As retrieving the data proved more challenging than initially thought, it was uncertain whether the process could be automated completely or only partly. On top of this, it was realized that involvement of a third party would have been required: every time there would be changes to the automation, the work would have to be bought outside the house. Moreover, coding work done for this specific environment would be hard to reuse in other processes. All of this combined, the estimated costs of the project started to escalate.

"So, the snowball was just growing constantly with AutoOp. First it felt good and clear: we have the software at use already and it's noteworthy. But when we went deeper and investigated the matter with various experts, we found a surprising amount of development that needed to be done, building interfaces, fetching information, and so on." (Service specialist, carrier services)

In the end, it was decided that the team would not proceed with *AutoOp*. Finally, RPA was found to be the best solution for automating the process. Prior to implementing RPA, Telco proceeded with an RPA-pilot to learn more about RPA's capabilities and training, running and costs of software robots. The team wanted to get first-hand experience about how software robots perform with systems used in the process: some systems are web based, and some have previously been a pain point of automation. Initially, the team also hoped to already gain capabilities to use the robots independently during the pilot. However, this proved to be outside the scope of the pilot. The conclusion was that developing own RPA capabilities is possible and necessary, but it takes more time.

People involved in the RPA-pilot included the project owner and the project manager, who were from the team that operates the process. Also, a representative of company IT was involved, taking care of matters related to system interfaces and arranging credentials for the external consultants. Corporate security took care of the required security clearances. External parties involved consultants from partner company, who configured the robots based on documentation and process walk-throughs with

Telco's team. The discussions in project steering group started during spring 2016, and in March, it was decided that a Proof of Concept (PoC) RPA-implementation would be made. At this point, also the partner company and platform were selected. After completing the security clearance process, the actual kick-off meeting was held in the end of May 2016. Ten person-days were reserved for configuring the robots, which was considered sufficient: this included a kick-off day held on-site, about seven days for teaching the robots and finally reporting for the PoC. PoC was built in June 2016, and the closing meeting was held in beginning of July. After the RPA-pilot was completed, implementation was put on hold while exploring other automation possibilities. In the end, RPA was found to be the best approach: after Telco started its RPA operations, the process was going into production in May 2017.

Telco chose RPA mainly because of the ease and short implementation time, as well as flexibility and adaptability of software robots, as they were able to operate in different environments. During the process, the project manager was forming a picture of the strengths and weaknesses of RPA. According to her, clear benefits were the easy integration of systems, and the possibility to automate also terminal phase processes where back-end integration is no longer a viable option. The possibility to add capacity to process quickly was also appreciated. The team also hoped to see improved quality and fewer errors in the future. This was partly based on the assumption that RPA would improve employees' work satisfaction, although it had been received with some caution:

"In the beginning it's scary, the change. It could be seen already during this pilot: there were careful questions coming from people about how this will affect the job descriptions. I had to tell them that, look, don't pack your things yet, but try to see this in the bigger picture: the primary goal is not to have less people, but to improve the quality of work. I believe that in the long run we can move expert work to tasks really requiring our input, like personal customer service. Personally, I hope that there are more opportunities than threats in the future." (Service specialist, carrier services)

4.1.3 Conclusion

This case demonstrated the selection between RPA and other automation approaches. The main selection criteria were associated with the number of required interfaces between the systems (as there was no existing interface between *OrderSys* and *NetInfo*), time to market (as it was considerably longer for *AutoOp* solution than for RPA), implementation costs (business case calculations) and in a way also the IT resource situation (in the sense that system development would have been required for both *OrderSys* and *NetInfo*).

4.2 Case 2: adding a new service to a customer's EntertainED subscription

The other automation case was conducted in Telco's Consumer unit. The automated process was adding a new entertainment service (called hereafter MyChannel) to customers' existing EntertainED subscription. EntertainED is Telco's subscription-based online entertainment service, where customers can stream, record and watch content of various TV channels and content producers using multiple devices.

Before automation, the process of adding MyChannel-service to customers' EntertainED subscription was outsourced to a partner. Therefore, Telco faced fixed costs associated with each MyChannel -purchase. Also, the process volume fluctuates quite a lot, as the demand for MyChannel is seasonal: time of the year and certain events influence the demand, and Telco sent monthly an estimate of the next month's demand to its outsourcing partner.

The desired outcome of automation was automating the manual processing of MyChannel subscriptions and having an end-to-end subscription automation for EntertainED users. As a result, customers' subscriptions would be delivered faster, and the number of human errors could also be reduced. Further, what comes to the objectives of the automation project, it was estimated to lower the data entry cost associated to MyChannel subscriptions.

4.2.1 Process before and after automation

When conducted manually, the process was as follows: The possibility of adding MyChannel-service to their EntertainED subscription is promoted to subscribers by the sales personnel of company called hereafter ChannelsInc. Once a day, ChannelsInc generates a list of customers who wished to add the service to their EntertainED subscription and delivers it to Telco. The list is a Microsoft Excel file, and it is sent as an e-mail attachment to a dedicated e-mail address, which directs it to a correct category in the ticket-system (a module of the CRM system for consumer customers). When the e-mail arrives, employees at Telco's outsourcing partner see that a new task has appeared and can start processing it.

Each row in the Excel contains information about customers and about the specific ChannelsInc product they wish to add to their subscription. The handler from Telco's outsourcing partner logs in to the ticket system and opens the attached Excel file. The handler then checks the contract number in Excel column D, and using the contract number, conducts a search in the CRM system to fetch the customer ID. The customer ID is then used to search *SubscriptionTool* in the CRM-system and to find Telco device assigned to the customer. The campaign and the name of the product that customer wishes to purchase are in Excel column K. In the *SubscriptionTool*, the handler selects the product from a list of products available for the customer's device. The order is placed and then processed by the CRM system.

A decision was taken to implement RPA to this process. After the implementation of RPA, the process was as follows: the user logs in to the CRM, after which the software robot does the remaining steps. In the ticket system, the robot selects the correct ticket category, opens the ticket and downloads the Excel-file. The robot then starts processing the Excel row by row. It gets contract information from Excel column D and moves to the search page on the CRM-system. The robot then uses contract information to find the customer ID from the CRM-system. After finding the customer ID, the robot moves on to the CRM *SubscriptionTool* and chooses the defined category and subcategory. It finds the service ordered by the customer from Excel column K and chooses the same service from the *SubscriptionTool* menu. Finally, the robot feeds the customer ID and correct service to the *SubscriptionTool* and moves the subscription forward. After this, the robot repeats the same procedure until all Excel rows are processed.

If the robot faces exceptions, it moves the order for manual handling. These exceptions include cases when text has been added to the comment field in a format that the robot does not understand. For example, if the sales person has agreed with the customer that order confirmation will be sent by letter, this information is added to the comment field and the robot will move the order for manual handling.

After the RPA PoC was ready, it was noticed that some modifications should be made to the Excel before moving the process into production. In the original Excel, the name of the product could be found in column K, but there was no product number corresponding to the one in Telco's system. This meant that ChannelsInc had to add a product number-column to the Excel they delivered to Telco. Also, in the production implementation, the robot was timed to process the orders on working days at 5pm. No manual involvement was, therefore, needed for starting the process. These modifications were made, and the process was successfully taken into production.

4.2.2 Decision

In this case, the decision between RPA and back-end system automation was clearer than in the previous case. Prior to implementing RPA, the team wanted to gain more understanding about RPA's capabilities, including how RPA manages the handling of data from different sources, and how much effort does automating a process with RPA really take. To compare the capabilities of different RPA platforms, the PoC was in this case built with three different platforms. One of the examined platforms was eventually chosen for production in Telco. The pilot was started on September 2016 with a demo session, where Telco employees demonstrated how the manual process was conducted. Consultants from a partner company recorded the process and built the RPA-implementation using about 24 person-days. When presented to Telco's team in November, the PoC was considered a success: RPA could perform all of the specified tasks. However, the team noticed that the process would still need some development before it could be taken into production. After setting up the RPA production environment, modifications were made to the process during April 2017. The process went into production on May 2017.

After the RPA-pilot was ready and proven successful, it was concluded that this process would likely have been automated sooner or later – but with what technique was harder to say. However, Telco's Service Manager in Order Management perceived that it was likely that there would have been challenges with traditional IT-development. First, the Excel and e-mail-based system for transferring the subscription data would have needed renewing. As the process involves receiving information from an external party, ChannelsInc, back-end system automation would have required modifying also the processes of ChannelsInc. This could bring along more uncertainties and dependencies, as completely controlling a process spanning over company borders is not possible. With RPA, modifications to the existing processes – apart from adding a column to the Excel – were not necessary.

"Probably we should have renewed the Excel-transmissions. We would have needed some other form of sending [data], and probably some form to fill, so it would then automatically 'swim' forward in IT. So, in a way, if we would have started this with 'normal IT', we should have renewed also the process of ChannelsInc." (Service manager, order handling)

Also, the process is still likely to develop in the future. According to a Telco employee involved in the project, process improvement work is never ready; a process should be continuously developed in a customer-centric direction. The interviewee envisioned that if RPA seems to work well, there is no showstopper for automating the process so that ChannelsInc could fill in orders right after talking with a customer, for example through a webpage. Orders would then be transferred to Telco's system and could be handled immediately, further reducing the time between order and delivery.

The pilot also made it apparent that process standardization is important in RPA. Telco employees involved in the pilot were positively surprised about the capabilities of RPA, and automation process was deemed relatively smooth. Due to positive experiences from the two described cases, next RPA candidates were already under mapping.

4.2.3 Conclusion

Interfaces between systems and changes in automation requirements influenced Telco's decision. It is likely that also process volumes affected the selection. Additionally, the fact that the process spanned over the company boundaries emerged as a factor in this case. Involvement of ChannelsInc caused additional challenge for back-end system automation, as also the process of ChannelsInc would have required renewing. Some modifications were needed also for RPA, but these were minor in comparison. Therefore, we may assume that RPA can reduce dependency on external parties when the automatable process spans over company's borders.

5 Discussion and Conclusion

5.1 Theoretical implications

We set out to investigate how a company makes the decision between lightweight IT and heavyweight IT. Drawing on a case study where both approaches were considered, we find support for the assumptions and findings on RPA selection criteria outlined in earlier studies (Fung, 2014; Lacity and Willcocks, 2016; Slaby, 2012). Compared to previous research, instead of merely addressing the potential transition from a manual process to RPA, our contribution lies in the articulation of how each of these criteria play out in the decision between the two different approaches to automation. Table 3 summarizes the findings from the two automation projects described above. RPA exhibits properties of a lightweight IT where it relies on multiple existing information systems and their frond-end interfaces (Bygstad, 2016). RPA uses the output of heavyweight IT, without modifying them. Our findings show that these properties also happen to be the core weakness of RPA when compared to heavyweight IT solutions.

Criterion	In favour of lightweight IT	In favour of heavyweight IT
Number of systems	Multiple	One
Process volume	Moderate to high	Very high
Stability of back-end system architecture	Changing	Stable
Stability of user interfaces	Stable	Changing
Interfaces between systems	No interfaces between systems	Systems have existing APIs
Time to market	Time critical	Not time critical
Permanence of process	Temporary	Permanent
IT resource allocation	Low	Medium to high

Table 3. Selection criteria for automation approaches

Our findings highlight the role of the multi-dimensional feature of stability of existing systems in choosing between RPA and back end automation. Interestingly, while the implementation of back-end system automation seems to require stable system architecture, it is not sensitive to interface changes on the presentation layer. Inversely, the implementation of RPA requires stable interfaces, but is not sensitive to system architecture changes.

5.2 Managerial implications

Our research also offers an insight for managers. Early RPA literature has focused on selection criteria such as volume of transactions, cost and cognitive requirements (e.g. Fung, 2014; Lacity and Willcocks, 2016). We build on these findings and focus on the features of system architecture and its fit to either heavyweight or lightweight automation. Based on our study, we claim that potential features that managers should investigate are the existing interfaces in their systems and their stability going forward. Thus, when considering the different alternatives for process automation, managers should carefully assess stability of their systems from multiple viewpoints. In line with the latest research (Hallikainen *et al.*, 2018), we suggest organizations to actively involve IT department in building RPA capabilities already at very early stage.

5.3 Limitations

While we had access to two rich empirical cases, out data is limited to two automation projects within one company operating in the telecom industry. Further studies could analyse automation projects across industries to investigate whether the choice is impacted by different factors in different contexts. Further research could also investigate cases where RPA was considered and rejected in favour of back-end system automation and contrast choice criteria with our findings.

Appendix – Coding scheme of the expert interviews

Theme	Code	#
Relationship between RPA and back-end system automation	RPA one tool in the automation toolkit	11
	Complementing, not competing	9
	Coordination with IT stakeholders important	8
	Other tools	7
	What can be done with RPA could be done with back-end system automation	4
	Same realities as in system development	4
	RPA requires well-functioning base systems	3
RPA strengths	Processes using multiple (silo) systems	24
	Speed	12
	Cost efficiency	10
	Integrating closed systems	7
	Modifiable/ lighter to change	7
	For small volumes	5
	Addressing seasonal demand	3
	IT pipeline full	3
RPA challenges	Change management	17
	Not as easy as promised (Need for specific capabilities, 11; Same realities as in system development, 4)	15
	Access rights & security	10
	Limited performance	8
	Not for big volumes or system reforms	7
	Complexity of processes	6
	Existing environment limits	2
	Limits of chosen RPA platform	1
Typical RPA cases	Temporary (Developing new services, 13; End of life cycle systems, 8; Waiting for back-end solution, 1)	22
	Systems or business rules still changing	7
	Reports or checks to support expert work	5
	Input or retrieve data from silo systems	5
	Process crosses company borders	2
Decision making criteria	Business case	8
	Time to market	5
	Scope	2
	Number of systems used	24
	Process volumes	13
	Anticipated system or business rule changes	11
	Interfaces between systems	6
	Temporary or long term	22
	Resource situation (IT pipeline full)	1

Table A1. Coding scheme

References

Aguirre, S. and Rodriguez, A. (2017). Automation of a Business Process Using Robotic Process Automation (RPA): A Case Study, In *Workshop on Engineering Applications*, pp. 65–71.

Asatiani, A. and Penttinen, E. (2016). Turning Robotic Process Automation into Commercial Success - Case OpusCapita, *Journal of Information Technology Teaching Cases* : 1–8.

Barnett, G. (2015). Robotic Process Automation: Adding to the Process Transformation Toolkit. Retrieved September 20, 2017, from https://www.blueprism.com/wpapers/robotic-process-automation-adding-process-transformation-toolkit

Bygstad, B. (2016). Generative innovation: a comparison of lightweight and heavyweight IT, *Journal of Information Technology* 32(2): 180–193.

Forrester. (2011). The role of IT in business-driven process automation. Retrieved September 20, 2017, from https://www.blueprism.com/wpapers/forrester-report-role-business-driven-process-automation

Fung, H. P. (2014). Criteria, Use Cases and Effects of Information Technology Process Automation (ITPA), *Advances in Robotic and Automation* 3(3): 1–11.

Hallikainen, P., Bekkhus, R. and Pan, S. L. (2018). How OpusCapita Used Internal RPA Capabilities to Offer Services to Clients, *MIS Quarterly Executive* 17(1): 41–52.

Institute for Robotic Process Automation. (2015). *Introduction To Robotic Process Automation*.

Lacity, M. and Willcocks, L. (2016). Robotic Process Automation at Telefónica O2, *MIS Quarterly Executive* 15(1): 21–35.

Lacity, M., Willcocks, L. and Craig, A. (2016). *Robotizing Global Financial Shared Services at Royal DSM*, *The Outsourcing Unit Working Research Paper Series*.

Manuel, P. D. and AlGhamdi, J. (2003). A data-centric design for n-tier architecture, *Information Sciences* 150(3–4): 195–206.

Mohapatra, S. (2009). *Business process automation*, PHI Learning Pvt. Ltd, New Delhi.

Prangnell, N. and Wright, D. (2015). *The robots are coming*, *A Deloitte Insight Report*.

Sabooniha, N., Toohey, D. and Lee, K. (2012). An evaluation of hospital information systems integration approaches, In *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*.

Serrano, N., Hernantes, J. and Gallardo, G. (2014). Service-oriented architecture and legacy systems, *IEEE Software* 31(5): 15–19.

Slaby, J. (2012). Robotic Automation Emerges As a Threat To Traditional Low-Cost Outsourcing, *HfS Research* : 1–18.

Sommerville, I., Cliff, D., Calinescu, R., Keen, J., Kelly, T., Kwiatkowska, M., Mcdermid, J. and Paige, R. (2012). Large-scale complex IT systems, *Communications of the ACM* 55(7): 71.

Willcocks, L., Lacity, M. and Craig, A. (2015a). *The IT Function and Robotic Process Automation*, *The Outsourcing Unit Working Research Paper Series*.

Willcocks, L., Lacity, M. and Craig, A. (2015b). *Robotic Process Automation at Xchanging*, *The Outsourcing Unit Working Paper Series*.

Yin, R. K. (2015). *Qualitative Research from Start to Finish*, Guilford Publications.