



UNIVERSITY OF GOTHENBURG

Email Analytics for Interaction Speed

Mining Organizational Latencies from Employee Inboxes

Bachelor of Science Thesis in Software Engineering and Management

KIRILL BLAZHKO

DMITRY IGOSHIN

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
Göteborg, Sweden, September 2013

The Author grants to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet.

The Author warrants that he/she is the author to the Work, and warrants that the Work does not contain text, pictures or other material that violates copyright law.

The Author shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Author has signed a copyright agreement with a third party regarding the Work, the Author warrants hereby that he/she has obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Email Analytics for Interaction Speed Mining Organizational Latencies from Employee Inboxes

Kirill Blazhko
Dmitry Igoshin

© Kirill Blazhko, September 2013.
© Dmitry Igoshin, September 2013.

Examiner: Ana Magazinius

University of Gothenburg
Chalmers University of Technology
Department of Computer Science and Engineering
SE-412 96 Göteborg
Sweden
Telephone + 46 (0)31-772 1000

Department of Computer Science and Engineering
Göteborg, Sweden September 2013

Email Analytics for Interaction Speed - Mining Organizational Latencies from Employee Inboxes

Kirill Blazhko, Dmitry Igoshin

University of Gothenburg

Gothenburg, Sweden

kirill@blazhko.org; dmitry.igoshin@gmail.com

ABSTRACT

Effective communication is one of the organization's success factors. Email is widely spread and often is the most used communication method, however some organizational latencies can happen which can lead to poor communication between employees and customers. The aim of this research is to find out if it is possible to reduce organizational latencies by improving email communication. The research was performed using grounded theory as both a method of data collection and data analysis, which was supported by the multiple case study research complemented by a number of interviews. The proof of concept research method was additionally used both to make the interventions in the interviews and as a complimentary way to support the grounded theory research process. As a part of the research results, we have discovered that in some organizations the reason for organizational latencies is that email is misused and that those organizational latencies can be mined from the employees' inboxes with the software prototype built during this research. We also make a proposal that in some organizations email should be replaced with an issue tracker (or a similar system).

KEYWORDS: email analysis, email analytics, organizational latencies, request handling, errand management, email analytics software, communication visualization software, email case study, business process latencies, email efficiency case study

1 INTRODUCTION

One of the important issues in medium and large organizations, especially those who handle a lot of customer inquiries – insurance companies, travel agencies, educational institutions – is to set up an effective internal and external communication strategy. Email is widely spread as a communication method, growing up to 3.375 billion accounts worldwide. As the time goes, mobile email usage is increasing rapidly; according to the recent research, 79% of smartphone owners use their device for email, and already 10% of consumers use a smartphone or tablet as their primary device for checking email (Adobe Systems Incorporated 2013, p. 3). Moreover, some findings show that email tends to replace phone calls: 48% of consumers prefer to take contact with organizations via email compared to 19% for phone (Kentico, 2013).

Email is considered to be a *memory bank* by some researchers from which one can retrieve and analyze emails from the organization's mail server (Nankani and Adarwal, 2007, p. 104). However, less attention has been paid to email efficiency as a business performance success factor (Joorabchi et al., 2011). Moreover, despite of wide use some researchers have questioned the efficiency of email as a communication channel and its effect on employee's working conditions (Derks and Bakker, 2010). However, there is a lack of research on the

topic of organizational latencies caused by email and the ways those latencies can be identified.

This research was initiated with a general hypothesis that by providing a simple, user-friendly way to analyze the email conversation flow, it is possible to help the employees to understand email processing problems (for example, providing a reply too late, involving not relevant colleagues in the conversation, forwarding the email message several times before it gets answered) and overcome them, improve organizational efficiency and response time, and increase the business performance. Therefore, the research questions addressed in this paper are as follows:

- *What are the reasons that cause organizational latencies and slow down customers' requests handling by employees in organizations that use email as a main communication channel?*
- *How can the reasons for organizational latencies be mined from the employees' email conversations?*

The purpose of the present study is to discover if it is possible to decrease the customer email request handling time in organizations. Request handling is a process of solving customer's request (reported via email, phone or other communication channel). Effective and fast request handling is one of the key success factors to achieve customer satisfaction (Gordon and McDougall, 2000). This research also evaluates the effect of an analytics software prototype – software for breaking a complex email conversation into smaller parts to gain a better understanding of it (Lapointe 2010), additionally using some metrics such as reply time to build statistical data – on the ability of organizations' employees to identify bottlenecks and time leaks which cause organizational latencies in their customers' email requests handling processes.

The approach used to answer the research questions was a mixed-method research: the grounded theory method supported by multiple case study research and by the proof of concept method. The grounded theory method was used both to drive the collection of the data together with the interviews and to analyze the collected data. This combination of different methods helped us to find answers to the research questions which are summarized in the Results section of this paper.

In this paper we want to show that the paradigm of an email message as a central part of a customer's request is a major driver for latencies that slow down the processing of customer's requests sent by email. Our contribution in the form of an analytics software prototype can be used to provide evidence to organizations that organizational latencies in their email communications do exist. As a conclusion, the research proposes to shift the focus from an email message as an entrance point to handling a customer's request to an autonomous request entity in a specialized issue tracking software. The developed software prototype is suggested to be used as a supportive tool in that shift.

2 METHODOLOGY

2.1 Methodology summary: mixed method

Because our objective was not only to discover problems with email communication, but also to propose how problems are to be solved, we used a mixed-method approach: the grounded theory method supported by multiple case study research and by the proof of concept method. This mixed-method approach is described in Figure 1.

The grounded theory method was used to discover the reasons that cause organizational latencies and slow down customers' requests handling by employees in organizations to answer the first research question (RQ1), while the proof of concept method was used to find the ways to mine those reasons from the employees' email conversations to answer the second research question (RQ2).

2.1.1 Multiple case study

Case studies are common in management research and also in software engineering research (Höst et al., 2012), therefore we considered the multiple case study method to be a good fit for our software-related research. The multiple case study can be an analysis of a set of persons, groups or events (Yin, 2009). Our multiple case study research was explanatory: we analyzed the data collected through the interviews to find the underlying principles (Shephard and Green, 2003), because using the grounded theory method we had no initial hypothesis for our research.

2.1.2 Grounded theory

The specific method employed to understand the problem was the grounded theory method. Grounded theory is a research method based on building a theory on data analysis results without having any initial hypothesis (Martin and Turner, 1986). By continuously iterating through the collected data we extracted a number of codes, categories and concepts which were used to form a theory explaining the causations of organizational latencies in the email communication process in organizations.

2.1.3 Initial interviews and observations

In order to collect a meaningful design science research knowledge base (Gregor and Hevner, 2013, p. 8) we have conducted multiple interviews with the employees of the organizations which base the working processes upon email interactions between stakeholders.

We followed the grounded theory concept which says that everything is data and thus the data collected through the interviews was not only transcripts of the interviews, but also the observations over the participants behaviour, their email usage patterns and their working environment (Glaser and Strauss, 1967).

2.1.4 Proof of concept and intervention

In parallel to the interviews which served us as a main engine of data collection and accumulation, we developed a software prototype which can analyse email inboxes and visualize the time that certain email conversations took to complete.

Additionally, as an element of mixed research methods we implemented an intervention: at the end of each initial interview we used our software prototype to analyse the email inbox of the interviewed person and to build visual timing diagram of his/her selected email conversations.

Our software prototype has helped us to analyse threads of email conversations by applying the graph theory and basic graph building and search algorithms on a given audience, counting involved parties as graph nodes and dynamically calculating the edges by evaluating communication email threads (Facebook, 2010).

What is important though, is that the intervention phase of the research was not very pervasive: it is the working employee who decided which email conversation threads would be analysed by our software prototype for organizational latencies and which email conversation threads could be considered as closed, i.e. have satisfied a customer request.

However, this had an unexpected advantage – one of the threats to the validity of the research, which was the choice of email conversation threads to analyse, was up to the employees who knew how to identify proper email conversations for an analysis.

2.1.5 Finalizing interviews

In order to properly reflect and learn from the performed research (Sein et al., 2011) we have pursued a set of finalizing interviews which addressed one question: can our software prototype be used to mine the reasons for organizational latencies from the employees' email conversations? The finalizing interviews extensively used the data collected through the observation and communication with the participants during the software prototype demonstration.

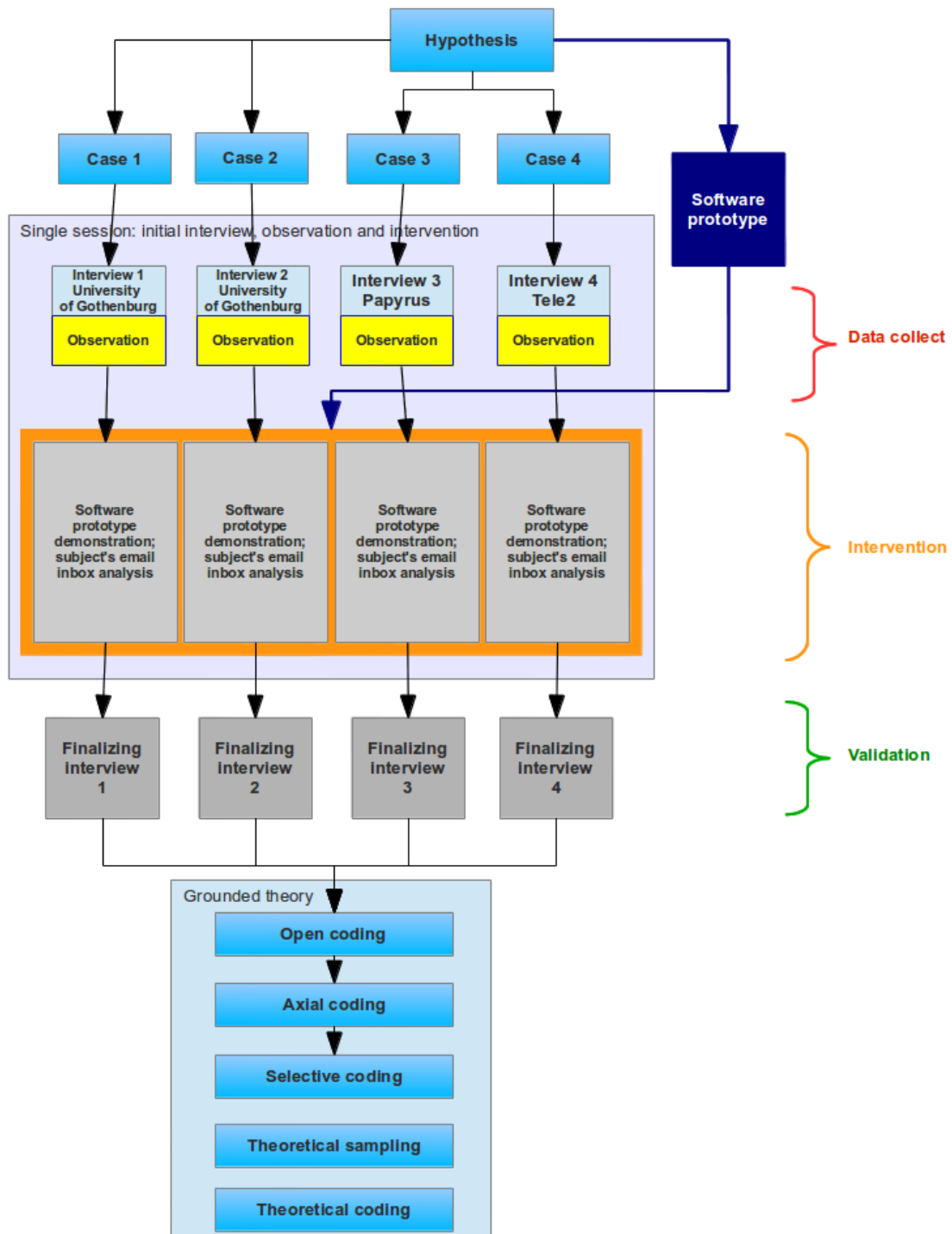


Figure 1 - Research process

2.2 Case selection and units of analysis

The selection of the research sites and interviewees, which is summarized in Table 1 was driven by the following factors:

- type of organization
- email volume of the interviewees which they get on the daily basis

As to the interviewee, he/she has to get at least 50 emails per day, be responsible for that communication with customers in some way and be aware of the whole conversation thread and responsibilities of involved people. It was also considered as an advantage if that person knows about organization's business processes and organizational structure.

Name of organization	University of Gothenburg	University of Gothenburg	Papyrus Sverige AB	Tele2
Type of organization	Educational institution	Educational institution	Paper production and complimentary services	Telecom operator
Department	Study Administration	Study Counselling	Software Development	Customer Support
Number of email requests per working day	>30	>10	>10	>200
Number of incoming emails per working day	>70	>60	>50	>200

Table 1 – case selection

2.3 Research sites

2.3.1 Research Site #1: University of Gothenburg

The University of Gothenburg provides study facilities to thousands of students, therefore there is a lot of administrative work to be done regarding the studies: course registrations, cancellations, scheduling and many other activities (University of Gothenburg, 2013). Most of this work is done by special units and departments of the university and most of the communications are done via email.

We have interviewed two employees in the study administration who process up to 200 emails per day with various student requests on a daily basis.

2.3.2 Research Site #2: Papyrus Sverige AB

Papyrus Sverige AB is a large corporation which has several successful projects in its portfolio. It is a leading supplier, supporting around 14000 customers on a daily basis and offering a wide range of products (Papyrus, 2013). Papyrus is paper, facility supplies and packaging merchant. The company also has a set of graphical and office paper brands, some of them provide recycled paper. Most of the software development is done internally in Papyrus.

We have chosen a project manager who manages a software development project in this company. Since in Papyrus all projects are the result of collaboration between different departments, the manager handles quite a significant number of email messages.

2.3.3 Research Site #3: Tele2 AB

In order to increase the variety of sources and cases to be analysed we added an end-customer oriented company to our research. Tele2 AB is such a company having millions of customers worldwide (Tele2, 2013). It offers services to both private people and organizations and has a large support department to deal with the subscribers. Tele2 services include 2G, 3G and 4G network access, television and landlines. The company has a lot of customer requests for a number of different services, and the email volume is quite large.

We have interviewed an ex-employee of Tele2's customer support department, who had been involved in handling a quite large number of customers requests.

2.4 Data collection

As shown in Figure 1, we collected the data through the interviews with people who process a lot of emails on a daily basis. All the interviews included observation and software demonstration (intervention). We also used those interviews to understand how email is used in the organizations and to collect feedback on the software prototype.

Open ended questions and unexpected detailed answers made us to decide on a face-to-face type of the interview: we doubted that we could get enough detail from an interviewee if we did the interview by email (Glaser and Strauss, 1967). Moreover, we preferred the face-to-face type of interviewing, because through it we could get access to observation of an employee's workplace and some administrative processes and protocols. This approach could reveal some earlier unexpected and unimagined data and could make obsolete some things that in the beginning we just took for granted (Neyland, 2008, p. 78).

Initial interviews (IR), Observation and Intervention

The initial interview was semi-structured and mainly consisted of open ended questions, because as interviewers we did not know in advance the IT technical proficiency level of the interviewees, their email habits, computer literacy level and protocols they followed when processing customer's requests by email. In addition, we aimed to make the interview to be exploratory to discover the issues perceived by the informants rather than us as researchers. The initial interview was structured as follows:

1. Initial interview (IR). The interviewee is asked the questions from the initial interview list.

Initial interview questions:

- How do you check your email?
- How often do you check your email?
- What do you usually do with an email message after you read it?
- How often do you use CC (Carbon Copy) function for your emails?
- How often do you forward emails to other people?
- Do you use your inbox as a list of tasks to be done (*To Do list*)?

2. Observation: We watch how an interviewee approaches his/her working place, what first he/she looks at in his/her email program and what other behavioral patterns can be observed.

3. Intervention: software demonstration and usage (INT). We install our software prototype (Outlook plugin) on interviewee's machine and ask him/her to choose an email thread (preferably problematic) to be analyzed. Our software prototype performs the analysis, and we present the result (diagrams) to the interviewee.

Finalizing interview (FI)

We provided the interviewees with the graphical results from INT so that they could have their time to analyze and reflect on that. After a few days we booked a finalizing interview with the interviewed and asked about the discovered organizational latencies and if he/she saw any improvements that could be done to the organization's communication process looking at the diagrams presented to him/her with our software prototype.

Finalizing interview questions:

- What reasons for organizational latencies can you see from the email inbox visualization diagram provided by our software?
- How these organizational latencies can be reduced?

2.5 Software: proof of concept

2.5.1 General development approach and requirements specification

Since our research methodology assumed that the software prototype was to be used as a tool for Intervention (INT) during our case studies, we analysed the general development approach before identifying the specific software requirements. Table 2 illustrates three main approaches we have evaluated.

Development approach	Motivation
Develop a standalone solution hooking into a user's mailbox	This way could provide us an opportunity to be independent and perform the analysis regardless of the email software the person has, but there are some issues: we would have to provide support for multiple protocols (at least, IMAP and MAPI)
Develop a standalone solution integrated with 3rd party mail services like Gmail via OAuth.	There is a need to support numerous services and there is a large probability that the organization uses its own mail server set up.
Develop a plugin for email client.	We found out that it is the most promising way, because some of the medium and large-scale organizations are running on their own mail servers and the employees access their mailboxes through desktop email client software, which we consider to be sufficient for our research.

Table 2 - development approaches

The software requirements specification process was driven by the research question two (RQ2) and by the data we collected through the interviews. RQ2 essentially requires the software prototype to be a discovery and a visualization tool which must be able to discover, visualize and help identifying the reasons for organizational latencies in the employees' email conversations. Those requirements are listed in Table 3. Req1 can be traced back to the research question two (RQ2) and it is a driver to the other software requirements.

Requirement ID	Requirement description	Requirement type
Req01	Provide a visualization diagram of a selected email conversation between a customer and the employee and his/her colleagues. This requirement drives all the other requirements.	Functional
Req02	Provide the functionality to manually choose message related to the subject. Depends on Req01	Functional
Req03	Integrate into one of the most popular email clients (Outlook). Depends on Req01	Functional
Req04	Display analytics results in the web browser. Depends on Req01	Functional
Req05	Save the analysis data under unique URL to be accessed later on. Depends on Req01	Functional
Req06	Provide analysis result within reasonable amount of time (up to 10 seconds)	Non-functional

Table 3 – Software prototype requirements

2.5.2 Email client choice

Another decision to take was to choose email client to integrate with. According to Litmus Email Analytics (Litmus Email Analytics, 2013), between 18 and 21.4% of email users have Microsoft Outlook as their email client software, and it is the most popular desktop-based one. Therefore our prototype was to be developed as a Microsoft Outlook plugin.

The client-server architectural model was chosen as for the following reasons: all the calculations are done on the server and the plugin is very lightweight, which provides us an opportunity to integrate with other platforms, such as Thunderbird, relatively easily.

2.5.3 Technology stack

We used a set of technologies (described in Table 4) to implement our prototype.

Component	Choice	Alternatives	Motivation
Application development language	Java	Ruby, Python, PHP	Good XML support, easy testable with the help of JUnit
Application framework	Spring	Play Framework, Ruby on Rails, Django	Partly driven by the language choice. We did not go for Play Framework because there is no support yet for deploying Play applications onto third party application servers.
Database	MySQL	PostgreSQL, Neo4j	For easier integration we went for SQL database. MySQL is sufficient for our purposes, given that there is no replication need yet.
Application server	Tomcat 7	Glassfish, Jboss	Tomcat is a lightweight server, convenient to use in development and easy to support. No features from Glassfish or JBoss are needed to implement our prototype.
Graph drawing library	Gephi	Sigma.js, mxGraph, Infoviz	Gephi has the support for GEXF (XML format for graphs)
Client-side drawing library	Highcharts	Raphaël, oCanvas, Dojo Toolkit, Bonsai	Highcharts is the one having good support for different diagrams.

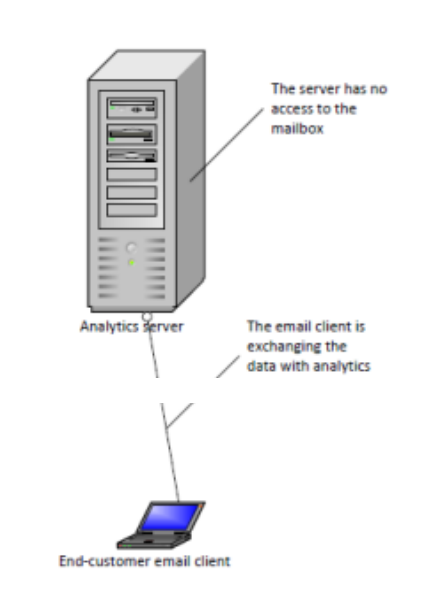
Table 4 – Technology stack

On the client side we used .NET Software Development Kit to develop a Microsoft Outlook plugin which is shown in Figure 4. That is the recommended way of the Microsoft Office plugins development. The server side of the solution is Java-based. It is a web application, having one XML endpoint for Outlook plugin communications and some pages to be accessed by end end-user. It is a quite usual MVC web application build on the Spring Framework, using MySQL as a database and Hibernate as a persistence provider. We have it hosted on Tomcat 7, which works behind a lightweight Nginx server (Nginx acts as a reverse proxy in our case). All the visualizations are done on the client side with the help of Highcharts and Gephi javascript libraries. The server is operated by the Ubuntu Linux Server Edition operating system.

2.5.4 Architecture

Given that the end customers might be using different email clients, and the analysis is done in the same way for all of them, client-server concept was a good choice for our prototype. This client-server concept is illustrated in Figure 2.

Figure 2 – System architecture



We use quite a standard way to present the analysis result (webpage in the browser), which gives us the possibility to develop server-side components in the most efficient way. We have developed an XML-based protocol, which is used to transfer data from an email client to the server. Therefore, several different email clients can be used to communicate with the same server.

As shown in Figure 3, the current prototype implementation supports only one message flow and does not provide advanced error handling.

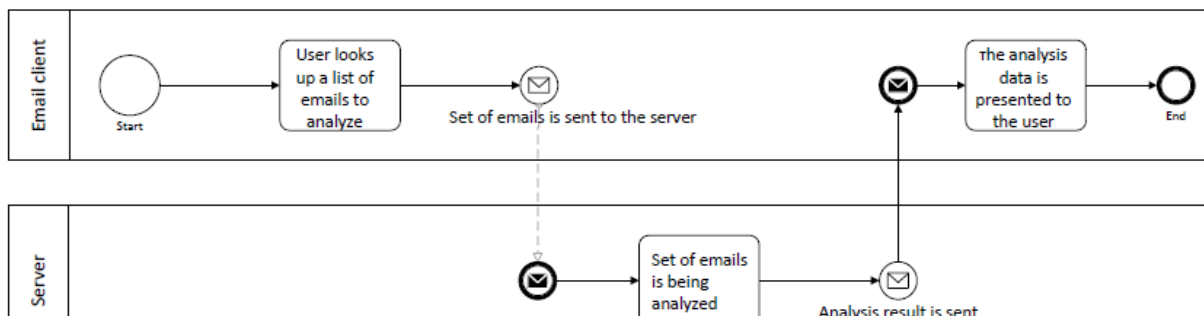


Figure 3 – Software prototype events flow

2.5.5 Visualization technology

One of the challenges we faced was the choice of visualization methods. We evaluated different types of flowcharts and noticed that it is not that much research available in the area of time-dependent data visualization (unfortunately, most of the methods we evaluated were target against static datasets). Finally, we went for a slightly modified swimlane diagram which is a way to show how a business task can be broken down into subtasks and in what order they need to be executed. It illustrates different functional capabilities of the elements of the architecture (Swimlane Guidelines, 2013)

We can also look at the email thread from a different perspective: instead of showing the relationship between email messages, as shown in Figure 5, we can visualize relationships between employees. That provides an opportunity to see generic communication patterns in an organization. Those relationships between employees and customers can also be visualized as a graph with nodes representing people and edges weighted differently depending on communication volume, as illustrated in Figure 6.

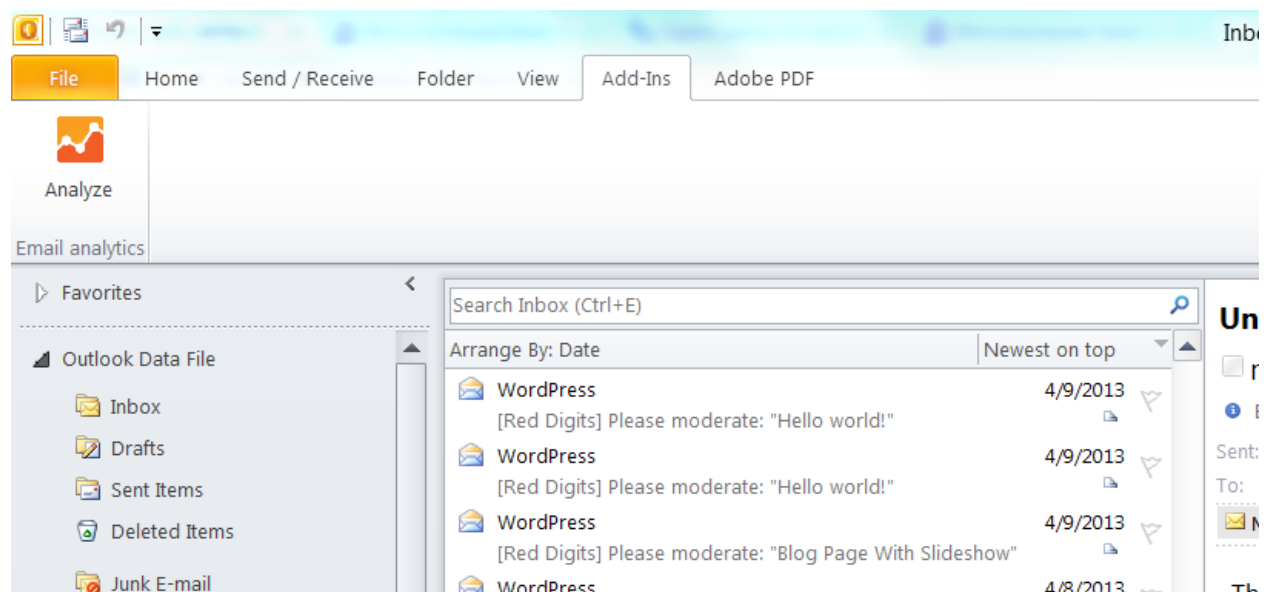


Figure 4 – Outlook plugin

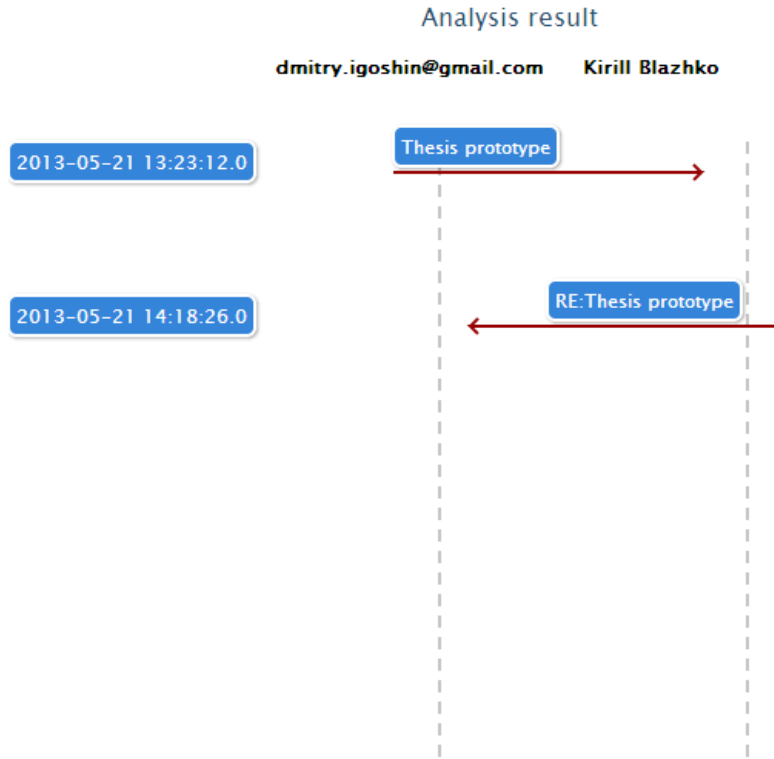


Figure 5 – Software prototype visualization example



Figure 6 – Social graph example

2.6 Data analysis: Grounded theory

We have extensively used the grounded theory research method to analyze the data that we had collected. The research process based on the grounded theory method is as follows:

1. The data was collected through literature review, interviews and interventions supported by the software prototype. We have tried to follow the concept of the grounded theory method which says that “everything is data” (Glaser and Strauss, 1967) and have written down not only the answers to the interviews, but all our observations. Furthermore, the *memoing* technique used at the open coding stage and the selective coding stage

(Strauss and Corbin, 1990) helped us to capture even the small details which we used to build and understand the emerging theory.

2. Key points were extracted from the collected field notes (this is called open coding). We pursued this routine task by hand, i.e. we read through each interview transcript and the set of notes and manually input the key points into a dedicated text file.

3. The key points were grouped together to form concepts, upon which one can build categories. We reviewed the key points a few times before we grouped them to make the concepts and then to derive the categories. This step also included a lot of discussion on how to organize and structure the collected data.

4. Axial coding was used to find conceptual relations between categories and to understand the basic framework of generic relationships (Strauss and Corbin, 1990). The axial coding technique was used to combine the variables in categories in different ways to sharpen the emerging theory (Strauss and Corbin, 1990, p.97).

5. Selective coding was used to find the central concept around which all the other categories revolved (Glaser and Strauss, 1967). Based on the previous steps we derived the central concept. We made a few iterations through the collected data to validate that this was exactly the central concept.

6. Next theoretical sampling was pursued, which is a process during which the collected data is sampled from the perspective of newly found central concept, which is also called the core variable. We looked at all the previous steps and the data from a new point of view driven by a newly established core variable. This approach helped us to define the emerging theory in a strict and formal way.

7. Theoretical coding is essentially a phase during which a hypothesis is built from the derived theoretical samples (Glaser, 1998): the theoretical samples were integrated together to form a theory which answered the research questions.

2.7 Validity

Validity is not considered to be a central issue in grounded theory (Glaser & Strauss, 1967), but instead such criteria as fit, relevance, workability and modifiability are the factors that a theory can be judged with (Glaser, 1978). Therefore, during the research we were concerned with how the theoretical framework we developed by applying the grounded theory method fits with the actual phenomena which exist in the studied area and how relevant it is to the real concern of the participants.

The workability of the theoretical framework was proven during the interview interventions and finalizing interviews when we found out that our theory was able to identify and explain the reasons that cause organizational latencies and slow down customers' requests handling by employees in organizations that use email as a main communication channel. This was proven by using the software prototype as a way of finding those reasons: when shown the visual charts generated by the software prototype from their email conversations, the participants were able to identify the causations of the latencies and they also confirmed that the concepts of the theoretical framework developed by us, such as *errand*, *errand status* and others, fitted into the real situation.

One threat to the validity, however, could be the modifiability of our theoretical framework. We have done four initial interviews and four finalizing interviews and there is some probability that, once added with new data (e.g. new interviews), the theoretical framework can be drawn less relevant to the studied incidents.

3 RESULTS

3.1 Literature research

As a first step in our research we have conducted a literature review. The review as a research activity serves the following purposes:

- Get a better understanding of the problem background and gain domain-specific knowledge
- Explore the approaches other researches have taken while addressing similar research questions
- Get an overview of techniques and methodologies that can be applied to this research

Different literature sources were used when doing the review such as Google Scholar (yielded 5,130,00 results for “email”, 53000 for “email analytics” with disabled “include patents” and “include citations”) and IEEE Xplore (returned 6,570 search results for “email”, 15 for “email analytics” and 106 for “email visualization”). We searched for papers on different topics in order to be able to look at our research questions from different angles. The main focus was put on finding and analyzing articles with similar research questions in order to identify how other researchers conducted their studies and what results they had gotten. We found two relevant papers, “EmailTime: visual analytics and statistics for temporal email” (Joorabchi et al., 2011) and “Visual Analytics for Organizational Email” (Nankani et al., 2007). However, the first one focuses mainly on analyzing email communication activity of the employees and finding general usage patterns and statistical data such as the number of sent and received messages in a given period of time, and the research is not addressing organizational latencies as such. The authors of the second paper consider email being the cause for organizational latencies, but their research focuses on internal communication between the employees, while we researched email as one of the main communication channels between an organization and an end customer.

The next step of our literature review was the search for methods and tools which were applicable for our research. During this process, we have found articles such as “Case Study Research in Software Engineering – Guidelines and Examples” (Runeson et al., 2012) which helped us better understand what research activities we needed to carry out.

The grounded theory method described in **2.6** helped us to collect data and to derive a theory from that data which helps to answer the RQ1, i.e. what reasons cause the organizational latencies in email request handling: there is a number of actions that an employee needs to perform in order to be able to handle a customer request. Performing those actions leads to the organizational latencies described in Table 6. The interventions that were performed during the interviews revealed the way that the organizational latencies reasons can be found (RQ2): with the help of the software prototype the interviewed employees can look at their daily email communications from a new point of view which shows them the causes of the latencies they usually experience.

3.1 Initial interviews, observations, interventions and finalizing interviews

We have collected various data about email request handling in each of the researched organizations and have found out that generally this process is cumbersome and complicated for the employees. With each initial interview (IR) we observed exactly the same pattern: each time an employee started to work and tried to decide which errands to process he/she read through the list of email messages located in his/her email Inbox folder. Of course, we noticed and recorded the fact that at least some of the messages were read through by the employee more than once. At this stage of the research we stated the fact that there was a reason for the organizational latency - a delay in organization's business process usually caused by flows in communication and information handling strategy (Vineyardsoft Corporation 2012): it was the redundant reading of email subjects, flags and read/unread statuses in the employees' email Inbox folders.

Moreover, the things got much more complicated once that responsible person responded to the employee: now an errand started to involve at least 3 persons (the customer, the employee and the other responsible employee). To understand the contents of the errand the employee in question had to read through all the email conversation from the very beginning.

This is where we have found the other evidence of organizational latency: when dealing with customers' requests (errands) each of the interviewed follows a certain sequence of steps:

- he/she tries to identify and recall the errand from the email message subject,
- then from the message body,
- and then from the whole email conversation.

Table 5 illustrates the main issues that we have discovered through Initial interviews, Observation and Intervention and then with Finalizing interviews.

	Case study 1	Case study 2	Case study 3	Case study 4
Research site	University of Gothenburg	University of Gothenburg	Papyrus	Tele2
Interviewee	Study administrator	Study counsellor	Project manager	Customer support manager
Initial interview results summary	Gets a lot of emails on a daily basis (approximately 70), answers some of them, forwards some to colleagues. <i>Uses important/not important flag as a status for messages (which request to deal with first), admits that it is easy to get lost in the process with such flagging and therefore to miss some requests.</i>	Gets a lot of emails on a daily basis (approximately 50), answers some of them, forwards some to colleagues. <i>Uses read/unread modes as a status for messages, says that such an ad hoc system is not sustainable.</i>	Gets emails (approximately 60) from colleagues, often from different departments, involved in many conversations as an observer. <i>Often sends email to multiple recipients, says that it is usually hard to divide responsibility for an incoming email request.</i>	Does not get emails from customers directly, but takes part in the discussion. Approximately 200 emails per day in the department. <i>Complains that it is often needed to read through all the email conversation to understand what the request was about and what should be done next and by whom.</i>
Software prototype intervention	Email is often forwarded to a colleague, and then the interviewee gets it back. For the current interviewee that was the program manager.	Many conversations are cyclic. The program manager was involved into too many conversations. Possible solution is to set up a list of frequent questions and answers both for internal and external use.	Often a person is on carbon copy on almost every mail, such as the project manager at the support department and the quality specialist, which can be replaced with weekly or monthly reviews.	Bottleneck: there is one responsible employee who has to have a look on every email to identify whom it should be assigned to
Finalizing interview	"Now I can see that I spend too much time understanding who is responsible for he errand"	"It seems that I spend too much time finding a right person for the request"	"I could save a lot of emails and time if I could see the status of each errand immediately and not reading each and every conversation over and over again"	"It would be much easier if the customer requests were automatically assigned to the right people"

Table 5 – Interviews

The most prominent observation was that all of the interviewees implied that their email inbox is essentially a place where all the errands were stored. When we asked them to show some of their email conversations which were related to some customers' requests, all of them did the following: they opened their email software on a computer, read through the list of all the email messages present in the Inbox folder and tried to identify an errand by viewing its read/unread status, flagged/not flagged status and even completed/uncompleted status (Microsoft Outlook allows to assign the completed/uncompleted status to an email message). Figure 7 illustrates that process of dealing of an employee with each email message in his/her daily work (*without our software prototype intervention*).

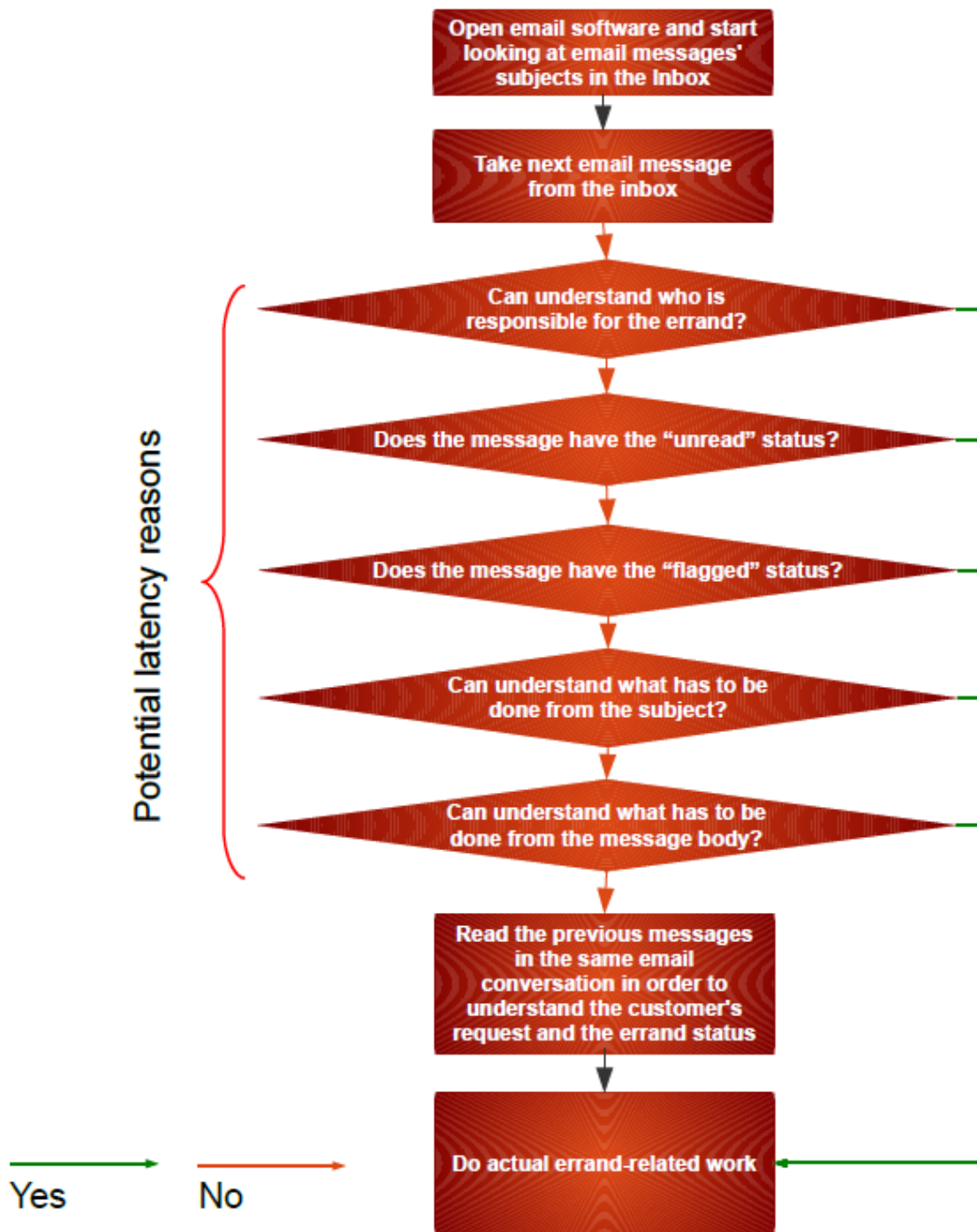


Figure 7 – Email workflow diagram

3.2 Discovered Concepts

We have coded and derived a number of codes, variables and concepts through the research. During the open coding phase we intentionally did not sort out any codes, whilst in

the axial coding phase we identified the core concept of the researched phenomena (Glaser and Strauss, 1967).

The central variable was the *request from a customer*, which also can be described as an *errand*. This is a basic social process around which revolves all the other variables of our research making for most of the variety in the studied cases. We also identified that this core variable is closely related to RQ1, because all the customer requests in the studied cases were delivered by email and thus there could be some organizational latencies when dealing with those customers' requests.

From the point of having an identified central concept we moved on to selective coding and started to selectively code the data we had collected. It was expected by us that two main units of analysis in our research turned to be a person and a request from a customer. The other variable which was mentioned frequently in the interviews was the concept of *responsible person* and *responsibility*. From the coded interview transcripts it seemed that it takes a lot of effort and communication messaging for an employee to identify a person who is responsible for an incoming errand and to forward the respective incoming email message to that person.

During the selective coding phase we pursued theoretical sampling and derived the theoretical concepts such as:

- *a request (an errand);*
- *a customer;*
- *a request status (affected by a responsible person);*
- *responsible person;*
- *a conversation.*

Next the theoretical codes emerged: we analyzed the relations between the identified concepts and discovered that the reasons for organizational latencies are caused by the fact that *the status* of an errand is separated from an email and is not stored and maintained anywhere. Figure 8 shows that separation and the relation of the derived theoretical framework to RQ1, i.e. the possible locations of organizational latencies in email conversations.

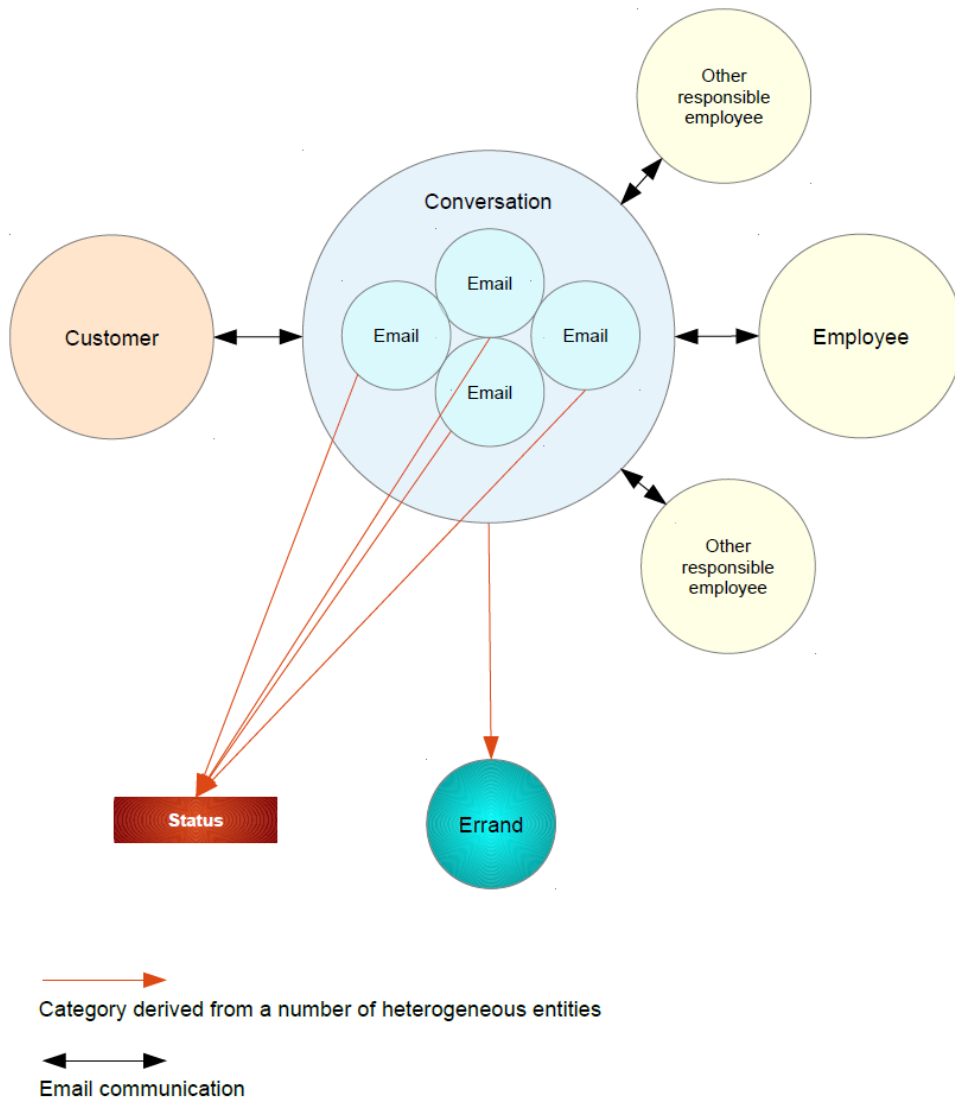


Figure 8 – Theoretical framework

Although central to the overall employee’s activity the interviewees have no tools to represent each errand as a single independent entity with a dynamic status attached to it. Instead, an employee has to deal with fragmented knowledge and data scattered through email messages, their subjects, flags and read/unread statuses, which increases the time needed to perform any operations with that errand. Essentially, at any given time a customer’s request data can be distributed among several people. What further adds to the organizational latencies is the fact that the interviewed employees often have to decide manually whom of their colleagues can be responsible for an incoming customer’s request. This information is usually not formally stated anywhere and is a part of the employee’s implicit knowledge.

Table 6 summarizes the discovered latencies reasons and serves as an *answer to the research question one (RQ1): “What are the reasons that cause organizational latencies and*

slow down customers' requests handling by employees in organizations that use email as a main communication channel?'

Reason ID	Raw data from the interview transcripts	Reason description	Is this information stored in the status of an errand?
Reason1	<i>Interviewee:</i> "I need to understand who is responsible for an errand"	A need to understand who is responsible for an errand.	No, the responsibility is derived from some undocumented implicit knowledge.
Reason2	<i>Interviewee:</i> "I need to check if a message has "unread" status to understand the actual status of an errand (active, finished, postponed, etc.)"	A need to check if a message has "unread" status to understand the actual status of an errand (active, finished, postponed, etc.)	No, some parts of the conversation can be marked as "unread" and the other can be marked as "read", which essentially makes it hard to decide if an errand is done or not.
Reason3	<i>Interviewee:</i> "I need to check if a message has a "flagged" status in order to sort out the less important errands."	A need to check if a message has a "flagged" status in order to sort out the less important errands.	No, some parts of the conversation can be marked as "flagged" and the other can be marked as "unflagged", which essentially makes it hard to decide if an errand is more important than the others.
Reason4	<i>Interviewee:</i> "It is often hard for me to understand what a request is about from the subject and sometimes from the body of a message."	Hard to understand what a request is about from the subject and sometimes from the body of a message.	No, the information is scattered through a number of email messages.
Reason5	<i>Interviewee:</i> "Usually it is difficult for me to understand what are the next actions by reading the latest message and I have to read the whole conversation thread again each time."	Difficult understand what are the next actions by reading the latest message; a need to read the whole conversation thread again each time.	No, the action to be done is decided by an employee based on reading through the whole conversation and some undocumented implicit knowledge.

Table 6 – Latency reasons

During the intervention phase of our research we have also studied that the interviewees found the diagrams generated by our software prototype to be easy to read and understand. All of them have agreed that the software prototype helped them to look at their email request handling processes from a new point of view and it helped them to identify the reasons for organizational latencies in the email errand processing. Therefore, we can answer the research question two (RQ2) in the following way: *the reasons for organizational latencies*

can be mined from the employees' email conversations through the use of an analytics software application developed as a proof of concept during this research by applying to it to the employees' email inboxes and by demonstrating them the visual charts generated from their emails' software analysis. As an example, figures 9 and 10 illustrate an email conversation related to two customer requests. Figure 9 depicts first errand via a swimlane diagram, while figure 10 represents another errand as a social graph.

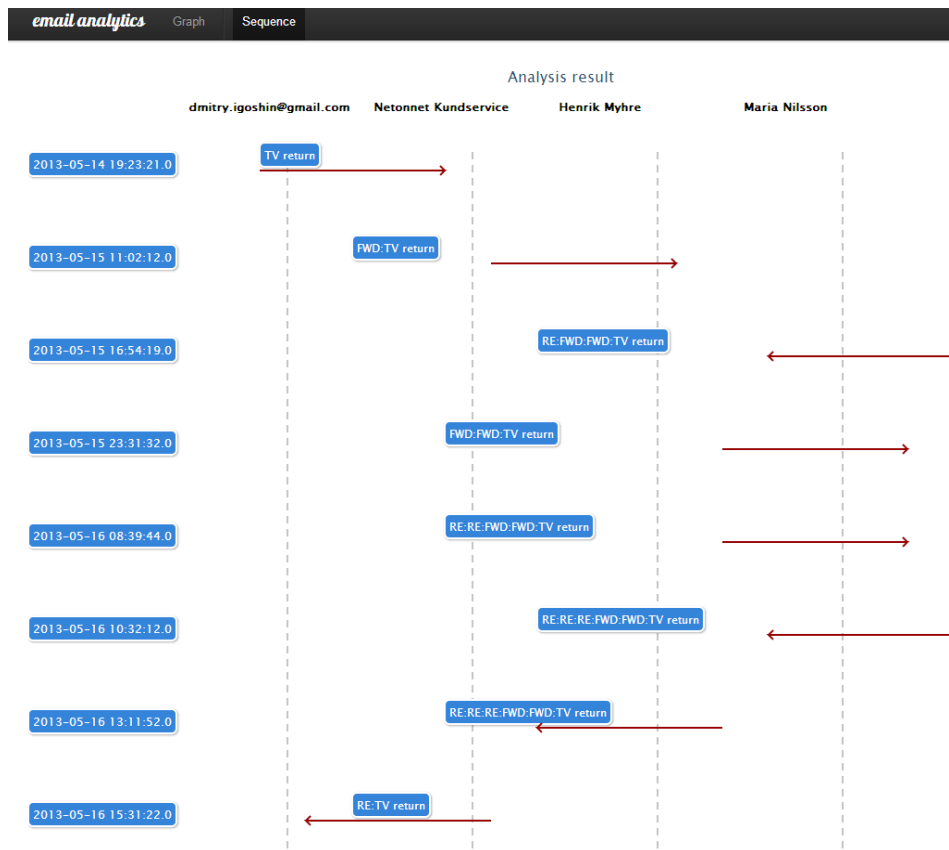


Figure 9 – Swim lane visualization

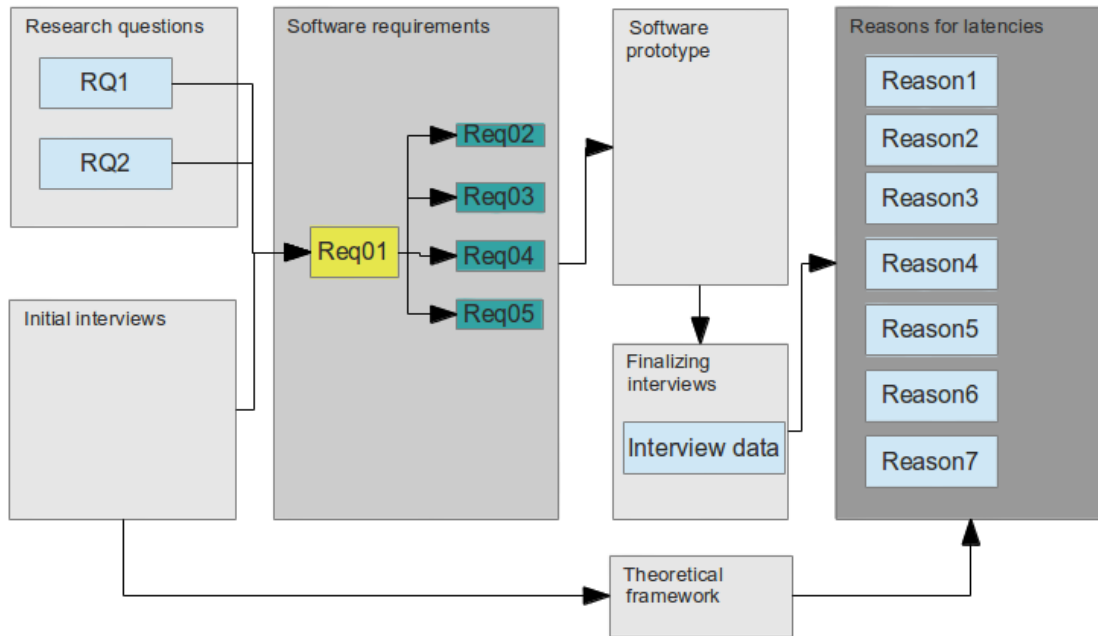


Figure 11 – Traceability diagram

We have found a number of answers to the Research question one (RQ1), i.e. the reasons for organizational latencies when handling customers' requests by email. They are summarized in Table 6. Working with these findings through our theoretical framework (shown in Figure 8) we have found out that the core variable in the emerged theory is the customer's request (errand) which has usually to be dealt with. The errand has two attributes: an owner (responsible person) and a status.

The organizational latencies described in Table 6 happen due to the fact that the employees try to use the email protocol as a medium to store a customer's request with its dynamically changing status and owner attributes: they have no place to store the information about the responsible person, the importance of a request, the next action to be done and the information about actual status of an errand (active, finished, postponed, etc.) This can be considered as an unexpected usage of a certain technology for wrong purposes: email is not a medium to store errands or issues, it is a method for exchanging digital messages (Crispin, 2003).

The software prototype developed during this research has helped us to answer the second research question (RQ2), i.e. the way of how organizational latencies can be mined from employees email conversations, and that has been proven by all the interviewees during the Finalizing interview (FI) sessions. Various visualization charts (Figure 9, Figure 10) generated by the prototype helped the employees to look at the request handling process from a different perspective and to discover and identify the reasons for organizational latencies in their email conversations.

What was discovered using the prototype is that customers' email requests are a big part of the employee's inbox, and those requests are traceable using analytics software such as the prototype we have implemented. What helped the interviewees to identify the reasons for organizational latencies was the ability of our prototype to visualize email threads with the use of various visual charts. This proved that reasons for organizational latencies (caused by lack of optimal email request handling) can be mined from employees' inboxes with the help of our analytics software.

In future, we can expect a larger variety of email analytics software applications. One of interesting ideas sprang from this research is an online service which can automatically analyze one's email inbox and reasonably quickly provide a set of reasons that cause latent times in that specific email inbox.

5 CONCLUSION

This research was performed to close the gap in the field of studying of organizational latencies in organizations which communicate to their customers by email. Provided with the data from a set of interviews, driven by the grounded theory method and supported by a software prototype the research identified the reasons for organizational latencies in the process of dealing with email customers' requests in organizations.

As a result, the research answered the question: "What are the reasons that cause organizational latencies and slow down customers' requests handling by employees in organizations that use email as a main communication channel?" The grounded theory method helped us to analyze the collected data and to build a theoretical framework which illustrates that when dealing with customers' requests by email, employees have no place to store the status information about a request, which leads to unnecessary work with each errand, such as identifying the responsible person, the actual status of an errand (active, finished, postponed, etc.), the importance of a request, the actual content of a request and the next action to be done for an errand. These unnecessary activities are the the reasons that cause organizational latencies and slow down customers' requests handling by employees in organizations that use email as a main communication channel.

This result was proven by demonstrating visual charts of email conversations analysis performed by the software prototype developed during the research to the persons who took part in the interviews during this research. The participants both identified the reasons for organizational latencies and confirmed that the software prototype is a way to discover the reasons for organizational latencies from their email conversations.

In order to remove the latent periods and decrease the time spent per errand by an employee, we suggest that an issue tracker system should be used instead of email protocol in order to deal with customers' requests. We believe that using email as a *memory bank* from which one can retrieve and analyze emails from the organization's mail server (Nankani and Adarwal, 2007, p. 104) is less effective than using an issue tracker system to handle email requests in an organization.

In such systems which are already used in some companies, email is used only to communicate updates information for a customer. All the interaction is done exclusively via a web interface to which the notion of an errand (also called request or issue) is central. The errand entity has the status and the owner attributes. Moreover, an issue tracker system can be set up in a way that allows automatic filtering of customers' requests and their assignment to respective responsible employees by the request subject. The diagram

illustrated in Figure 12 shows the concept of an issue tracker system depicted through our theoretical framework which is demonstrated in Figure 8.

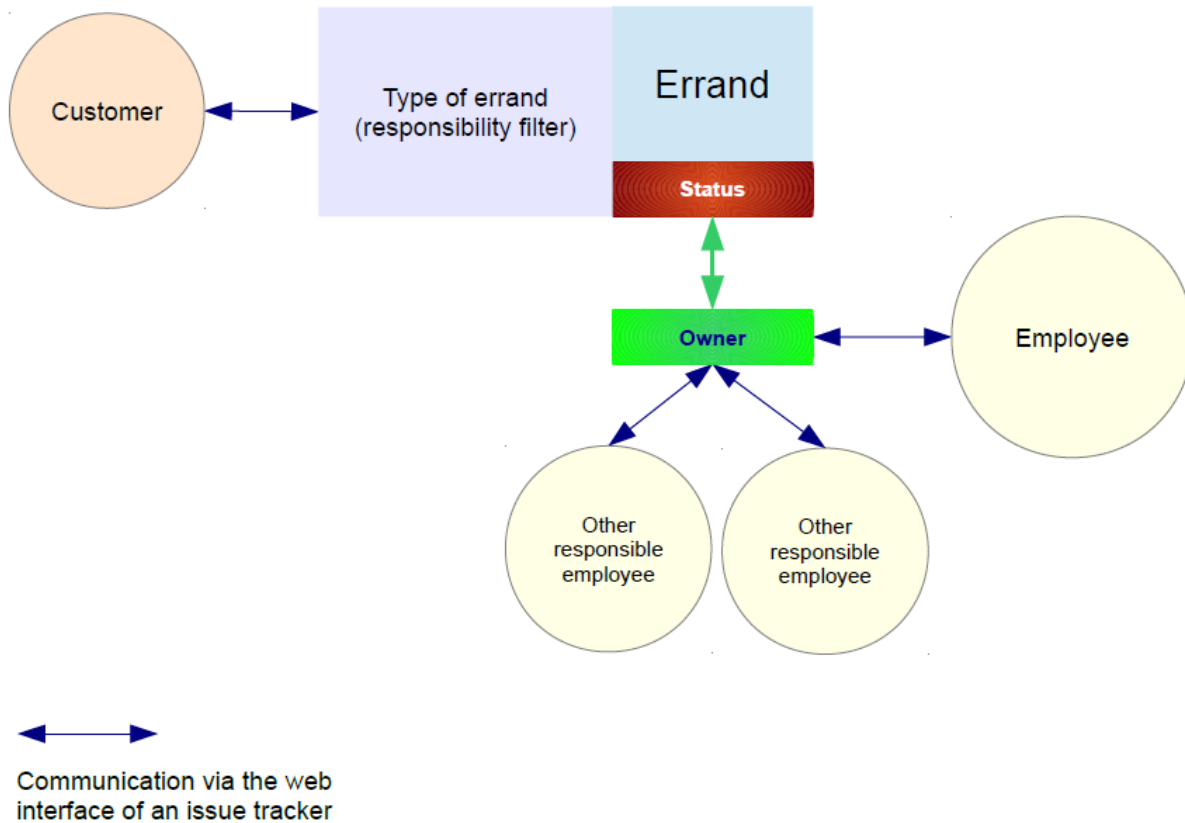


Figure 12 – Issue tracker concept

To motivate the shift from the email-based request handling to the issue-tracker request handling in organizations we suggest using the software application which we developed during this research. It can mine, discover and help identifying the reasons for organizational latencies and thus provide clear evidence of the need to change the working process of an organization to the organization's management. Since a decision to introduce some new working process in an organization can be influenced by various factors like budgeting, cost of implementation, learning curve and others, we believe that our software application can be used as a proof-generation tool to influence such a decision.

6 ACKNOWLEDGEMENTS

We would like to express our gratitude to our supervisor Lars Pareto for the useful comments, remarks and engagement through the learning process of this bachelor thesis.

Furthermore we would like to thank Dina Koutsikouri for introducing us to the art of technical writing and for the support on the way.

Also, we like to thank our interviewees: Study Counsellor Suzana Plancak, Study Administrator Anna Södergård, an ex-employee of Tele2 and a project manager of Papyrus for providing the data from industry.

We would like to thank our classmates for their extremely valuable feedback.

7 LICENSE

This work is licensed under the Creative Commons Attribution - No Derivative Works 2.5 License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/2.5/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA. Contact the authors to request other uses if necessary.

8 TRADEMARKS AND SERVICE MARKS

All trademarks, service marks, logos and company names mentioned in this work are property of their respective owner. They are protected under trademark law and unfair competition law.

9 DOCUMENT USAGE

This work is optimized for both screen and paper use. For environmental reasons it is recommended to use the digital version where applicable.

REFERENCES

- [1] Adobe Systems Incorporated. (2013). 2013 Digital Publishing Report: Retail Apps & Buying Habits. Available: <http://success.adobe.com/en/na/programs/products/dps/1301-28915-mobile-retail-survey.html>. Last accessed 20th Apr 2013.
- [2] Daantje Derks, Arnold B. Bakker. (2010). The Impact of E-mail Communication on Organizational Life. *Cyberpsychology: Journal of Psychosocial Research on Cyberspace*, 4(1), article 1.
- [3] Ekta Nankani, Abhishek Agarwal. (2007). Visual Analytics For Organizational Email . International Business Informatics Challenge and Conference 2007.
- [4] Glaser, Barney G. and Strauss, Anselm L. (1967) The discovery of grounded theory: strategies for qualitative research. Chicago.: Aldine.
- [5] Gordon H.G. McDougall, Terrence Levesque, (2000) "Customer satisfaction with services: putting perceived value into the equation", *Journal of Services Marketing*, Vol. 14 Iss: 5, pp.392 - 410
- [6] Gregor, Shirley; Hevner, Alan. 2013. "Positioning and Presenting Design Science Research for Maximum Impact," *MIS Quarterly* Vol. 37 No. 2, pp. 1-XX
- [7] Jack P. Friedman (2012). *Dictionary of Business Terms*. : Barron's Educational Series.
- [8] Mino Erfani Joorabchia, Ji-Dong Yima, Christopher D. Shaw. (2007). EmailTime: Visual Analytics and Statistics for Temporal Email. *Visualization and Data Analysis 2011*, Volume 7868, article id. 78680Q, 11 pp. (2011).

- [9] Kentico Software. (2013). Kentico Survey: Websites Second Only to Word of Mouth in Driving Brand Affinity. Available: <http://www.kentico.com/Company/Press-Center/2013/Kentico-Survey-Websites-Second-Only-to-Word-of-Mou>. Last accessed 2nd Apr 2013.
- [10] Evan Lapointe. (2010). A Better Definition of Web Analytics. Available: <http://www.atlantaanalytics.com/practicing-web-analytics/a-better-definition-of-web-analytics/>. Last accessed 24th Apr 2013.
- [11] Litmus Email Analytics. (2013). Email Client Market Share. Available: <http://emailclientmarketshare.com/>. Last accessed 24th Apr 2013.
- [12] Sheila Mello (2003). Customer-centric Product Definition: The Key to Great Product Development. -: PDC Professional Publishing. 256.
- [13] Neyland, D (ed.) 2008, Organizational Ethnography, SAGE Publications Ltd
- [14] Papyrus (2013), About Papyrus. Available: <http://papyrus.com>. Last accessed 24th May 2013.
- [15] Shepard, Jon; Robert W. Greene (2003). Sociology and You. Ohio: Glencoe McGraw-Hill.
- [16] Robert K. Yin. (2009), Case Study Research: Design and Methods. Fourth Edition. SAGE Publications.
- [17] "Facebook: One Social Graph to Rule Them All?". CBS News. Retrieved July 11, 2010.
- [18] Per Runeson, Martin Höst, Rainer Austen, Björn Regnell (2012). Case Study Research in Software Engineering – Guidelines and Examples. Wiley.
- [19] Glaser, B, (1967). Theoretical Sensitivity: Advances in the methodology of Grounded Theory. Sociology Press.
- [20] Patricia Yancey Martin & Barry A. Turner, (1986). Grounded Theory and Organizational Research. The Journal of Applied Behavioral Science. vol. 22, no. 2
- [21] Sein, Maung K.; Henfridsson, Ola; Purao, Sandeep; Rossi, Matti; and Lindgren, Rikard. 2011. "Action Design Research" MIS Quarterly, (35: 1) pp.37-56.
- [22] Tele2. (2013). Om Tele2. Available: <http://www.tele2.se/kundservice/kontakt/om-tele2.aspx>. Last accessed 24th Apr 2013.
- [23] Agile Modeling. Swimlane Guidelines.(2013). Available: <http://www.agilemodeling.com/style/activityDiagram.htm#Swimlanes>
- [24] University of Gothenburg. Available: <http://gu.se>. Last accessed 24th May 2013.
- [25] Vineyardsoft Corporation (2012). The Six Steps to Reducing Business Latencies. Available: <http://www.northbrookconsulting.com/The%206%20steps%20to%20reducing%20Business%20Latencies.pdf>. Last accessed 19th Apr 2013.