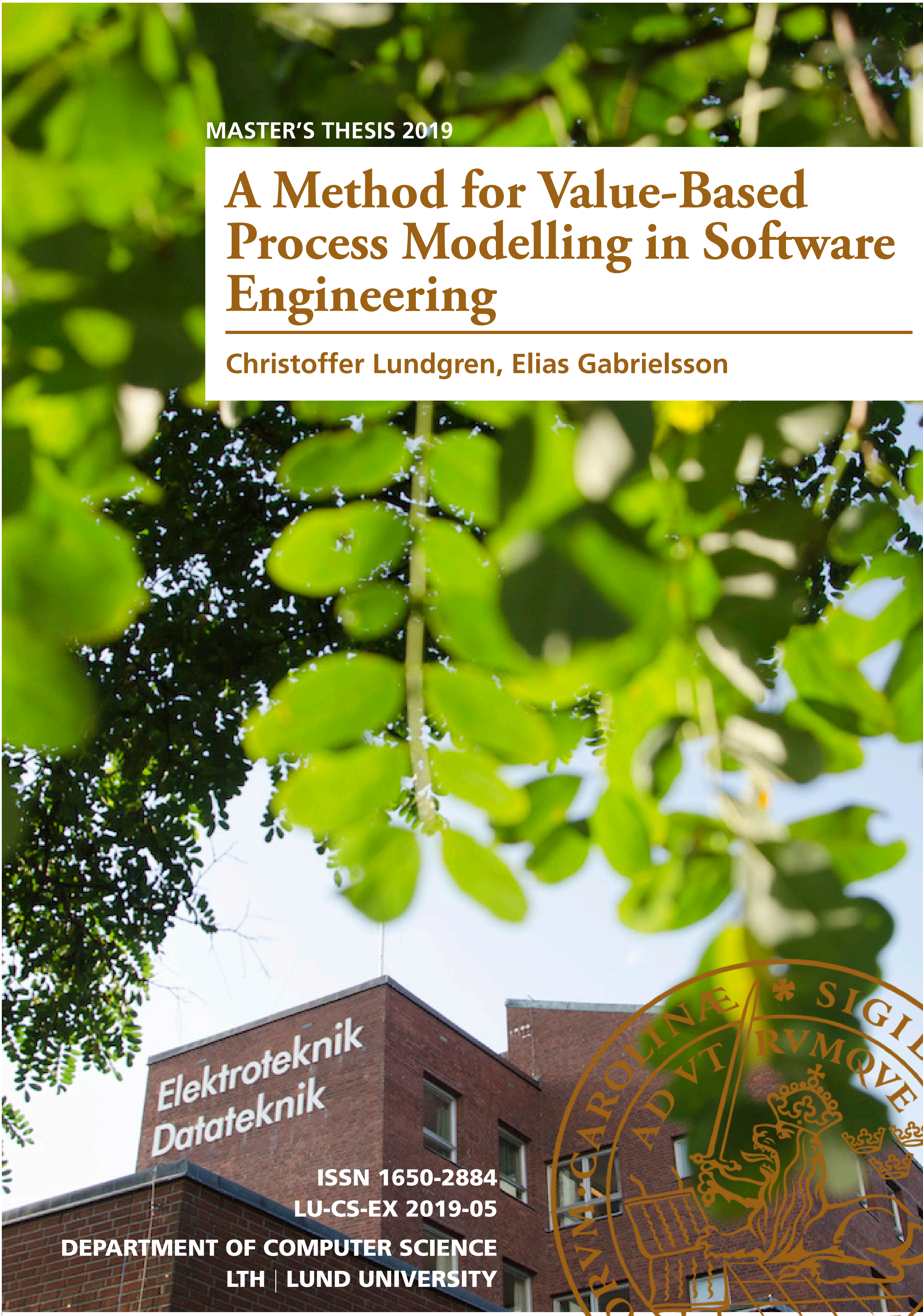
MASTER'S THESIS 2019

A Method for Value-Based Process Modelling in Software Engineering

Christoffer Lundgren, Elias Gabrielsson



ISSN 1650-2884
LU-CS-EX 2019-05

DEPARTMENT OF COMPUTER SCIENCE
LTH | LUND UNIVERSITY

EXAMENSARBETE
Datavetenskap

LU-CS-EX: 2019-05

A Method for Value-Based Process Modelling in Software Engineering

Christoffer Lundgren, Elias Gabrielsson

A Method for Value-Based Process Modelling in Software Engineering

(An Industrial Case Study Using Design Science Research)

Christoffer Lundgren
mail@christofferlundgren.se

Elias Gabrielsson
elias@gabson.se

May 28, 2019

Master's thesis work carried out at Combitech AB.

Supervisors: Fredrik Dahlstrand, fredrik.dahlstrand@combitech.se
Elizabeth Bjarnason, elizabeth.bjarnason@cs.lth.se

Examiner: Per Runeson, per.runeson@cs.lth.se

Abstract

Background
For consulting companies within the software industry is it necessary to be able to adapt to their clients' varying workflows and organisational structures. To be successful in this adaptation is it central for consultancies to have flexibility in their internal organisations to allow them to quickly adapt to new business cases.

Purpose
Combitech AB needed an overview of their organisation to be used as a management tool to allow them to be flexible while maintaining repeatability of service production in the software development field.

Method
We applied a design science research approach to build and evaluate a solution to the addressed industry problem. We iteratively designed the ECPM Method and Notation Language, and evaluated these artifacts by applying them to Combitech AB. When the ECPM Method and Notation Language were applied to Combitech AB, a process-oriented overview in the form of a value-based process map was generated.

Results
The main results of the study were a process map describing Combitech AB, and the evaluated ECPM ("Elias Christoffer Process Mapping") Method and Notation Language. The method, used to collect and analyse data, consisted of 26 activities. The notation language, used to draw the process map, comprised 20 graphical components and 10 rules to illustrate value flows.

Conclusion
According to the evaluation at Combitech AB did the value-based process map help them resolve their organisational difficulties.

Keywords: Process Mapping, Process Notation Language, Value-Based Process Mapping

Acknowledgements

We would like to thank Elizabeth Bjarnason for her supervision of this thesis.

We would also like to thank Fredrik Dahlstrand for his supervision, and everyone else at Combitech AB for their contributions to this thesis.

Contents

Chapter 1
Introduction

The focus of this thesis was to design a method and a notation language for value-based process modelling, and to apply these two to Combitech AB in order to create a process model of their workflow.

This chapter presents the background to the industry problem in Section 1.1, followed by an introduction to the studied case company in Section 1.2. In Section 1.3 the scope of the thesis and research questions are described. Finally, in Section 1.4 the disposition of this thesis is outlined.

1.1 Problem Background

Repeatability is a desired property during software development. This does not only include the technical aspect, e.g. reproducablity of test results and builds, but also the capability to repeat similar work in new contexts, for example developing software for a new car series based on the efforts in the previous one.

Repeatably in terms of the organisation of human capital, i.e. *structural capital*, is important for a software consulting company as that structure is the origin of their sellable value. A highly developed structural capital is for example central in the ability to swiftly incorporate new team members and stakeholders in projects while maintaining the capability to produce a complex technical product.

Producing software is not made easier by the fact that the methodologies used by organisations for carrying out work are widely diverged and complex, sometimes making it hard to know what exactly contributes to the end result. This is natural as software often integrates into many other industries which all have their unique technical needs and particular management structures. Software engineering in itself also has a variety of disciplines which all imposes their specific practices and methods; IEEE presents fifteen major sub-disciplines within the software industry in the work *"Guide to the Software Engineering Body of Knowledge"*[1].

Since year 2000, iterative workflows have been widely adopted in software organisations. This paradigm shift was not fully compatible with traditional management methods used in organisations as the time horizon under which planning and decision making differed. Being able to connect the established management methods with the iterative workflows effectively and efficiently is therefore a difficult but desirable capability for an organisation to have.

A general trend for companies in almost all industries is the transformation from being vertically integrated, i.e. the company creates all value in-house, to instead being specialised within a particular area and focused on their core business [2]. This transformation is highly applicable to software organisations as they often provide a specialised product within a field or technical framework, e.g. Mobile Development, Web, Embedded etc.

Companies that delivers these services needs to dynamically integrate their own workflow with their customers', but still maintain a core workflow ready to be adapted to other clients or projects. In order to react to change from a consultant organisational viewpoint, scalability in combination with above mention repeatably are good traits.

Some of the problem discussed above is not new in any sense, but the unique properties of software development have created a need for better suited methods. Already in the early days of the modern manufacturing industry *Scientific Management* emerged as a method to measure productivity [3]. Many methods and perspectives have since been developed, and central in several of these is to model the world as *processes* which takes an input and transforms it to an output. Processes and process modelling have been used extensively in this thesis work.

1.2 Case Company: Combitech AB

In order to contribute to the solving of the general problems presented in the previous section, the authors performed a case study at Combitech AB.

Combitech AB, henceforth Combitech, is a Nordic technical consulting company within the security group Saab AB. The revenue in 2017 was 2.3 billion SEK with approximately 1 800 employees over all offices. Combitech is currently growing both organically and through acquisitions. The clients are found within the manufacturing, service, and defence industries, as well as within the public sector. The company has presence in more than 30 locations, of which the office in Malmö, Sweden was studied in this thesis [4].

The office in Malmö had seen an increased interest from clients to buy innovation-driven software development as a service. This differed from the traditional consultancy services that Combitech had offered; from previously mostly have been working in the clients' organisations to instead carry out the development within Combitech's own organisation. This also meant that Combitech now owned the software development processes to a larger extent than previously.

Combitech therefore needed a better overview of how value was created, and a way of creating, updating and customising such an overview. The overview specifically needed to relate traditional management methods to contemporary software development frameworks.

For more information regarding the projects which were studied at Combitech, please refer to *Section 4.4: Case Data*.

1.3 Scope and Research Questions

The authors presumed that the issue could be resolved by creating an overview in the form of a process model which illustrates the value-creating activities in the organisation. The reason behind selecting the perspective of value-creation is twofold; *value creation* can act as a common denominator for various fields of work, and the value-creation perspective enables management procedures to be flexible in order to lead creative work such as modern software development.

The approach is interdisciplinary in the sense that it merges knowledge from the fields of software engineering and logistics. Understanding how management methodologies specific to the software industry and process management originating from logistics relates to each other is a key topic for this thesis.

The research questions specify the scope of the authors' approach to the industrial problems and specifically to the studied case, and concerns both *how* to create the process map, and *what* it may look like:

RQ1: How could the work of creating a process model for an organisation that develops software be organised efficiently?

RQ2: What can a value-based process model for an organisation that develops software look like and specifically which entities of information should be included?

RQ3: Which properties of a process model could be reconfigurable to ensure higher versatility?

This thesis was delimited to the study of a selection of at the time active projects at Combitech's office in Malmö, Sweden.

1.4 Disposition

In *Chapter 2: Background* the fundamentals of software development processes and process management are introduced. In *Chapter 3: Related Work* the process concept is expanded further as value-based process modelling, different ways to illustrate processes, and value within software are discussed. In *Chapter 4: Research Method* the research method used for this thesis is described together with details of the studied case at Combitech. *Chapter 5: Results: The ECPM Method and Notation Language* contains the designed method and notation language, used for creating the process model. *Chapter 6: Results: Applying ECPM at Combitech* describes the results, mainly the process map, from the execution of ECPM at Combitech. *Chapter 7: Results: Evaluation of ECPM* presents the results from the evaluation of ECPM. In *Chapter 8: Discussion* the relevant empirical data are discussed in the light of related work. Finally, in *Chapter 9: Conclusions* this thesis is concluded and the research questions are answered.

Chapter 2
Background

Recurrent when discussing management throughout several disciplines today is the occurrence of the term *process*. When searching within literature on the subject a united definition does not seem to exist. In a structured literature study of the phrase "process management" performed by Palmberg [5] 77 full articles were condensed. From the results six properties which could be seen in the majority of all definitions of the term *process* were identified:

- "*Input* initiate the process, *Output* delivers the result"
- "Interrelated activities"
- "Horizontal: intra-functional or cross-functional"
- "Meet purpose or value for customers and stakeholders"
- "The use of resources"
- "Repeatability"

Palmberg also created a wide definition of *process* from all the articles read:

> "A net process definition can be condensed to: A horizontal sequence of activities that transforms an input (need) to an output (result) to meet the needs of customers or stakeholders."

In exploration of approaches and tools to manage and improve processes, i.e. *Process Management*, Palmberg distinguished two distinct directions within the literature:

1. **Process management for single process improvement:** A structured systematic approach to analyse and continually improve the process.
2. **Process management for system improvement:** A holistic manner to manage all aspects of the business and a valuable perspective in determining organisational effectiveness.

Improvements are context specific and covers a wide range of topic such as standardisation, definitions of measures, homogenisation within an organisation and similar. These two directions are both relevant in the areas of *Business Process Management* and *Software Development Processes*, presented in Section 2.1 and 2.2 respectively, as these two terms are used throughout this thesis.

2.1 Business Process Management

Traditionally companies have organised and observed themselves hierarchically, commonly by function or division. Functions have delivered some specific capabilities to their organisation or worked with certain tasks, while divisions grouped functions together around a business field, product or geographical area, according to Edgren and Skärvad [6].

With this organisational viewpoint a software company can for example be divided into sales, development units, testing, etc. An organisational chart based on functions is shown in Figure 2.1.

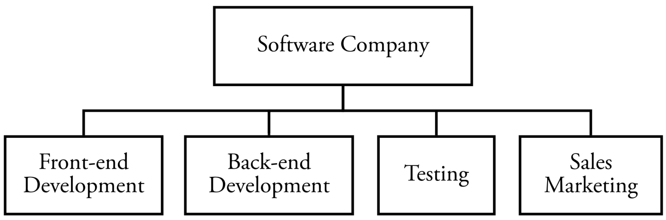


Figure 2.1: An example of a functional organisational chart.

This traditional way of thinking of an organisation was criticised because of its inefficiency to make functions to work together towards shared goals. A lot of last decades' complex products and systems need tight integration between functions to succeed. To counter this flaw with functional organisation a new perspective on business organisation emerged as the *process paradigm* which was popularised in the early 90's. Davenport [7] concludes this perspective with:

> "While we cannot measure or improve hierarchical structure in any absolute sense, processes have cost, time, output quality, and customer satisfaction. When we reduce cost or increase customer satisfaction, we have bettered the process itself."

The concept of value, which flows between internal as well as external processes, is central in business process management. Rummler and Brache [8] specifies that by observing a hierarchical organisation as a series of processes, with value flowing between them, is it easier to identify *who*, as in which function or role, add value to *what*.

Business processes are often illustrated as a box, which represent the value adding activity, with two arrows representing the input and the refined output, as in Figure 2.2.

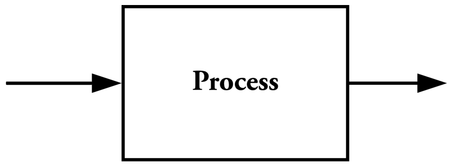


Figure 2.2: An example of an illustration of a general process

Multiple processes can be connected together and form a chain of value-creating processes, illustrated in Figure 2.3. As the output from one process is the input to the next is it possible to trace value-flows backwards and forwards.

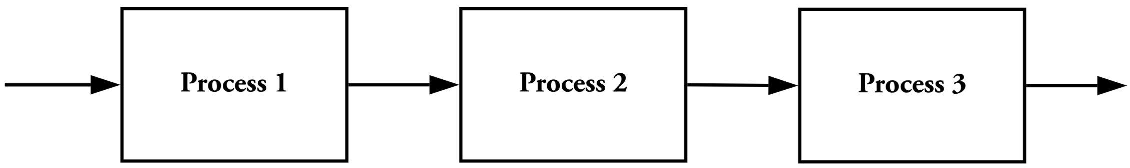


Figure 2.3: An example of an illustration of a value chain

The sole term *value* can be perceived somewhat vague and unspecific to be practically useful. To give guidance in this, Rosing et al. [9] have developed three major categories of processes depending on which business value a process produces. Using these categories while working with processes could enhance communication and focus efforts in multi-stakeholder contexts as it unites agendas.

Main Processes or *value processes* is where the fundamental value is created. Usually these processes can be identified by backtracking the output, which is delivered to an external customer. In the software context development activities can be regarded as main processes.

Management Processes are required to lead the main process, for instance activities like planning, budgeting and monitoring. These are also known as *control processes*. *Scrum* is an example of a software management process [10].

Support Processes provides main processes with everything they need to reach their objective. Examples are a services like maintaining a hardware test bed or staffing of a project.

Rosing's work [9] presented above was published in 2014 and hence *not* included in Palmberg's literature review from 2009 [5]. The two authors' publications support each other as both derived three very similar process categories; while Rosing presented the above three, Palmberg found "Operational delivery processes", "Strategic management processes" and "Supportive administrative processes".

2.1.1 Common Challenges with Process Management

Ljungberg [11] points out that business process management is a way of thinking and not just a method to model organisations. To fully gain the advantages that are made available through the use of process management this thinking must penetrate the entire organisation.

This makes implementation of process management related to and dependent on the field of *change management* as it gives guidance on how to transform organisations. An example of a recognised change management model is Kotter's eight step model [12]. However, change management is beyond the scope of this thesis.

Another pitfall recognised in the 90's when process management grew popular was that while organisations did reorganise to process management, they did not always focus on the *correct* processes. Keen called this the *process paradox* [13]. In his research he sorted processes according to meaningfulness, here presented in descending order starting with the most important one:

Identity Processes are unique processes which distinguishes the business from competitors and the general market. There are similarities with Rosing's definition of *Main processes*.

Prioritised Processes are processes which determine how efficiently the identity processes can be run.

Background Processes are processes which deliver necessary support for the organisation in general. These have large similarities with Rosings's *Support Processes*.

Enforced Processes are processes required by regulations and law.

Traditional Processes are processes driven by old habits and behaviours which have lost their former purposes.

Awareness of the process paradox will have important implications when developing methods for how to create process models, which is one of the research questions.

2.1.2 Process Measurement

In opposite to the holistic view, where the connections between processes are in focus, process measurement is more about improving individual processes' characteristics in order to contribute to the big picture.

Measurement is possible to be applied onto different aspects, both directly and indirectly. Ljungberg suggests that measurement can be divided between *process/activity measures* and *resource measures* [14]. The first one manifests into the measure related to the process itself, how it is carried out, qualities on output or input, etc. The second one concerns the resources the process needs to be conducted. Measures could be resource consumption, resource utilisation, and resource characteristics.

Specifically for software Fenton and Bieman [15] present a third category, *Product Measures*, covering metrics of the produced deliverables. Fenton and Bieman also argue that it is possible to use different product measures based on context, i.e. if the product is used in its designed environment or observed in isolation.

Ljungberg also argues that measures targets too much on data that gives too little information about the future, such as how much a process have consumed or used. Furthermore, a lot of process data are summarised and simplified where only mean values are taken into consideration and variations are ignored [11].

2.1.3 Sequential and Iterative Workflows

An additional viewpoint of process management, which has been discussed more in the last twenty years, is the use and modelling of iterative patterns and methods. Having iterative patterns is not something new. In manufacturing this can be traced back to Toyota's continuous improvement concept *Kaizen*, incubated in Japan's buildup after World War II. This process is also summarised by "Plan", "Do", "Check", "Act", PDCA, which influenced forthcoming ideas within manufacturing such as the highly data driven improvement method Six Sigma. In manufacturing a planned and sequential workflow is still dominant, though.

When software development started as a discipline of its own it inherited management practices from manufacturing. Royce created the sequential *Waterfall-model* in 1970 which gives a good overview of development activities but has questionable performance track record, as many instances tend to not follow his advice to iterate [16].

In early 2000 there was a movement among software developers to counter what many considered to be an inflexible way of producing software. Software developers sought after a more responsive way of managing development which quickly could react on new insights and requirements. The thoughts was summarised in the *Manifesto for Agile Software Development* with the purpose of challenging the established way of software development:

> "Individuals and interactions over processes and tools
> Working software over comprehensive documentation
> Customer collaboration over contract negotiation
> Responding to change over following a plan" [17]

When using these agile principles in practice the workflow tends to be cyclic and iterative and therefore the name *iterative methods* is also common.

The manifest mentions a shift from processes and it is worth to consider what that implies. A core thought of the manifest is to facilitate *react to change*, something a stiff and detailed process description could hinder. On the other hand if a process description has a high level focus with flexibility to let the person working with process self choose and adapt work methods, the authors does not see any contradictions between these approaches.

In the space of software development there is a smorgasbord of frameworks utilising parts of or the entire agile manifest as core values. Common to all is that they are iterative. Examples are *Scrum* [10], *eXtreme Programming (XP)* [18] and *Kanban* [19].

The scope of what the frameworks defines varies. Scrum defines a wide range of product oriented roles, meetings and artifacts compared to Kanban which circulates around task visualisation and control within a team. The methodology eXtreme Programming focuses on having a lean workflow with a minimal number of non-code artifacts and continuous communication with the client to achieve high responsiveness.

2.1.4 Standardisation and Certification

Standards and frameworks have evolved in order to give guidance on how to glue different parts of process management together. This was needed as process management interacts with and involves methods, tools and concepts which often spans over a variety of functions and organisational levels and thereby is complex and hard to grasp.

Some of these standards have corresponding certifications that organisations could qualify for which could give credibility and negotiation advantages. Sometimes compliance to a standard is not just a negotiation advantage, but also a requirement when participating in public procurements.

One focus of certifications is to describe the process capabilities an organisation has achieved in order to indirectly measure an organisation's performance. A notable example in the software field is the *Capability Maturity Model Integration (CMMI)* [20], originally developed out of an initiative from the US Department of Defence in order to improve selection of subcontractors.

There are also standards whose purpose are to guide and to be a reference on how process management shall be implemented. They touch the same subjects as for example *CMMI*, but are more instructive and less concept theoretical. An example of a standard oriented towards service delivery is *The Information Technology Infrastructure Library (ITIL)* [21].

Process management is also a central part in the ISO 9000/9001 standard series [22, 23], which primarily addresses quality management. The series focuses attention on how quality aspects from organisations can be improved and measured by applying a process management viewpoint.

Agile methods' relation to process management has also been in focus for standardisation. In the last decade a lot of organisations especially in the software field have had challenges scaling up agile practices throughout their entire organisation. That challenge lead to the creation of a new sort of frameworks dealing with integration of agile frameworks into existing organisations structures. Examples are *Scaled Agile Framework (SAFe)* [24] and *Disciplined Agile Delivery (DAD)* [25].

2.2 Software Development Processes

The field of software has since its inception developed a specific set of processes and activities to improve the creation of software. This section will introduce a brief overview of influencing software processes based on the work *"Guide to the Software Engineering Body of Knowledge"* [1].

Palmberg's defining summary of processes is still very applicable even if the field of view is narrowed to *software* processes.

2.2.1 Requirements Management

Development of software does usually involve a wide range of stakeholders, external as well as internal, from different industries. Formalising and managing all these stakeholders' needs is the work of requirements management. Requirement management could be seen as a way of mitigating the risk of building and producing the wrong software.

Lauesen proposes that activities for requirement management can be sorted in three main categories based on their purpose: elicitation, validation and verification [26]. As they addresses topics that concerns the value of the entire life cycle can they be found both in the start and the end of a process based value chain.

2.2.2 Design and Implementation

The most obvious activity for software development is the production of source code artifacts. This process is usually carried out in two steps; first, an architecture phase where a system design is produced from a set of requirements and second, an implementation phase where the desired functionality using the design from the first part is implemented in code. These phases can either be seamlessly integrated into each other or very formally separated depending on the project scope.

How implementation, i.e. coding, is carried out can also vary between occasions; from support with well defined specifications to only having loose design directions. Depending on the competence level of the workforce is this not necessarily a bad thing but a desirable property to leave room for creativity.

2.2.3 Testing

The process of testing plays an important role within software development. Testing aims to identify defects in the software artifacts and assess the quality of the software. Sources of error can be split up in five major origins according to Burnstein [27]: *Education*, e.g. the developer's lack of competence. *Communication*, e.g. information have not reached the recipients who needs it. *Oversight*, e.g. things got omitted to be performed. *Transcription*, e.g. mistakes were done during the creation. *Process*, e.g. the way things were performed is misleading the efforts.

Testing itself should be followed up by classification and triage processes which are interfunctional. Test results can for example lead to re-prioritisation of resources which a requirements process is accountable for. In order to catalyse the administrative process a wide range of test databases and test repositories are available and often used.

2.2.4 Configuration and Deployment

For software to be useful, business-specific adaption is often needed. This adaption is usually performed at deployment and does often regard parameterisation and configuration of a software artifact which originally was made with a generic focus. For example, the software powering a generic telecommunications system is probably configured to conform to a country's regulations before deployment.

Even though software has characteristics of being theoretically easy to duplicate, are there high costs associated with doing so as the maintenance costs grows with the duplication. According to Babich three typical problems of maintaining software exists: maintaining identical software copies, modifying shared data and updating data simultaneously [28].

Milligan, with experience of working with software configuration management at IBM, concluded seven desirable key attributes providing support and business value into these issues:

safety, stability, control, auditability, reproducibility, traceability and scalability of software [29].

Configuration management processes tries to solves these issues by imposing methods like change control boards, release management, continuous integration, etc. onto the organisation. Specific software toolsets such as *GIT* and *Subversion* assists these methods.

2.2.5 Operations

After a software component has been deployed into production for a customer, the focus of the delivering organisation shifts from development to providing operational value for the customer. This usually includes tasks such as monitoring, system migration, system maintenance, backup and recovery.

One aspect which not touches the software directly, but still is important during operations, is on-site training for the staff who use the system. This is less common for customer facing product but is common within enterprise solutions.

Chapter 3
Related Work

The use of processes in the proposed solution to the studied problem have the properties of *"Meet purpose or value for customers and stakeholders"* and *"Repeatability"* according to the definitions by Palmberg introduced earlier in *Chapter 2: Background*.

This chapter presents relevant work with these two process properties as its base to contribute to the answering of our research questions. Section 3.1 concerns others' work within describing value-creating processes with regards of semantics, visualisation and elicitation. The following Section 3.2 summarises relevant definitions of what *value* means within the software industry. Finally, in Section 3.3 the relationship between streams of information and streams of value are discussed.

3.1 Value-Focused Process Modelling

There are several process models that describes value creation. Ljungberg experiences that almost all of the models use a graphical representation, something Moody considers to be an efficient way of communicating this kind of information [30, 11].

The authors of this thesis have identified two fundamental design choices which dictates the possible use cases of the process model.

The first design choice is to either directly model recursive and circular work patterns, or to instead indirectly model them by illustrating the refinement of a produced service or goods. When designing with a recursive pattern, models needs to be read more like a flowchart in opposite to non-recursive models which can be read as a value-refinement chain. Models which uses a recursive approach tend to be more detailed and impose a heavier cognitive information load on the reader. This is apparent already on a trivial level when comparing Figure 3.1, illustrating a model with recursive elements, and Figure 3.2, depicting a model with value-refinement focus:

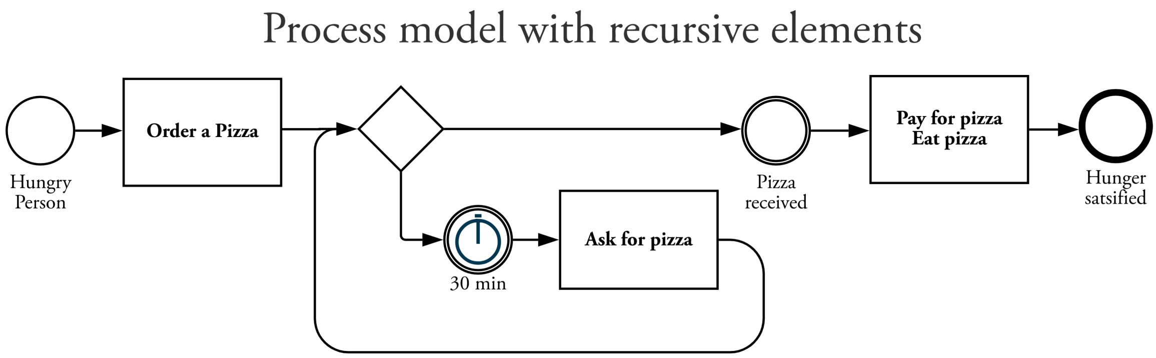


Figure 3.1: Trivial example concerning pizza serving using BMPN 2.0 modelling language, a standard that uses recursive patterns. Here drawn according to Briol [31].

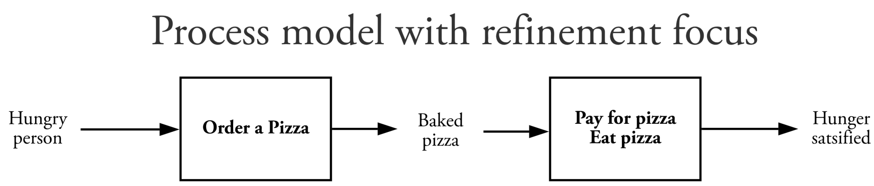


Figure 3.2: A generic process visualisation of the pizza example also drawn in Figure 3.1. Please note the considerably less amount of visual clutter even on this trivial level when the recursive loops are removed.

The modelling style seen in Figure 3.2 also shows the situation of pizza serving but with refinement focus. The reality which the model claim to describe does of course also contain repeating elements, such as the customer's *"Ask for pizza"* task; the difference is the focus on *refined value*. All things that leave a process contains a transformed input with refined, i.e. higher, value for the customer. Value-based process models will therefore, unlike the recursive model in Figure 3.1, not illustrate for instance wait time, which is just a delay and not a perceived value for the customer.

The second design choice concerns the process itself; if the process shows *why* or *how* something should be done. Depending on which level of detail the models seeks to visualise will the description of the processes vary. Ljungberg and Larsson points out that directing processes' descriptions towards explaining *how* things should be done, i.e. in *instructional* terms, in favour of *why*, i.e. in terms of *purpose and intention*, will have a negative trade-off on the flexibility of development [11]. A rule of thumb in formulating a good process description with a *why-focus* is that a process description should include a *verb* with a *noun*, optionally accompanied by an *adjective* that emphasises the uniqueness of the process.

The competence of experienced staff could be better utilised if they have freedom to choose their own working methods (*why-focus*). This mindset is well acknowledged and implemented in a variety of organisations dependent on initiative and creativity for success. On the other hand, strict control with checklists and and similar can reduce complexity in well defined problem spaces (*how-focus*). This principle is used both for experts, e.g. pilots, and to drastically improve performance of lay-mans, e.g. voluntary manpower. As can be seen is this closely tangible to the leadership and culture of the organisations.

Closely related to process descriptions is how processes are controlled. As introduced earlier in *Section 2.1: Business Process Management*, control processes, or management processes, exist to plan and realise main processes. They therefore often constrain *how* realisation can be done. Consider the example in Figure 3.2. It is not unreasonable to think that food laws will dictate requirements that the process of creating a pizza needs to comply with and hence also affect *how* it will be realised.

According to Lehman and others, who conducted studies at IBM's software units, is it suitable to use different control processes depending on the project's maturity, the availability of requirements, and domain awareness. Lehman with others also found it successful to monitor projects' state of requirement awareness and adjust control processes accordingly [32].

3.1.1 Visualising and Interpreting Process Models

When applying a process approach to improve the understanding of the customers' needs, artifacts that illustrate the value-delivering organisation will likely be produced, commonly referred to as *process maps*. A process map is created for two objectives according to Ljungberg [11]: to provide understanding and to support realisation. A process map can target of both these objectives or just one of them.

Many modelling languages used for drawing process maps take inspiration and can be traced back to the modelling language *Structured Analysis and Design Technique* developed in the late 60's by Douglas T. Ross. The work was later republished in the more commonly known form of IDEF0, short for "ICAM DEFinition for Function Modelling", where ICAM in turn is an acronym for "Integrated Computer Aided Manufacturing", [33] which was a set of knowledge series for including computers in the manufacturing process for the U.S. Air Force.

IDEF0 introduces among other things the semantics of *control* and *support* to the basic process modelling block previously introduced in Section 2.1. The extended process modelling block is illustrated in Figure 3.3.

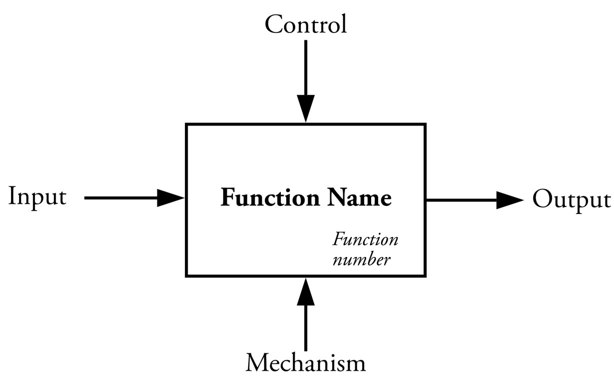


Figure 3.3: IDEF0 buliding block. *Mechanism* is also referred to as *Resources* or *Support*.

Rosing's three kinds of processes: "Main", "Management", and "Support", also described in *Section 2.1: Business Process Management*, can be visualised by the use of IDEF0.

This has in turn been used to form modelling languages such as *Value Process Modelling* [14]. Modelling languages used for mapping communication within software teams like FLOW by Schneider et al. [34] have also utilised IDEF0 as a baseline.

Some modelling languages uses "swimlanes" to categorise processes. A "swimlane", or "functional band", in the process context refers to a certain swinlane-like area of a process map to which processes assigned to a specific function or role are drawn. Ljungberg with others have noticed that this influences organisations to go from a process way of thinking into a traditional functional mindset. Ljungberg's rationale behind this is that functional silos are explicitly drawn into the modelling language akin to organisation charts and therefore blocks the process way of thinking [14]. Practical experiences indicates that modelling languages using swimlanes often induce heavy cognitive load as they produces many relationship that crosses each other graphically [35].

A useful concept that swimlanes use is the sense of direction. Moody recommends that maturation of information, in this case value, is drawn in one spatial direction to give sense of refinement, i.e. the more to the right in the map, the higher the value [30].

It is important that a process map involves and engages the associated stakeholders in order to build up commitment to actually use it. Introducing a process map introduces a new mental model for the stakeholders and usually also involves some changes of work routines. This will take time to grasp for the involved stakeholders. Lundgren working in the field have experienced that a key success factor for stakeholders to ingest a process map is to include parts of the stakeholders' current mental models to allow them to feel in control and recognise themselves [35].

Another way of including stakeholders is to visualise their participation in different processes. It is also possible to generalise this by using *roles* instead of *persons* in the process map, which gives the benefit of removing dependencies to a specific person.

One way of including stakeholders is the *Responsibility Assignment Matrix* concluded in PMBOK® [36] which defines the responsibilities that each role shall fulfil for different processes. The matrix can define different sets of responsibilities. A common configuration is the *RACI Matrix*, which is an abbreviation of "Responsible", "Accountable","Consulted" and "Informed". An example of a RACI-matrix is shown in Table 3.1. Yang et al. and Khan et al. have shown that the use of RACI-matrices could be an efficient tool for management to assign and delegate responsibilities in software related projects [37, 38].

	Role I	**Role II**	**Role III**	**Role IV**
Process A	A	R	I	I
Process B	A			R
Process C	I	A, R	C	I

Table 3.1: An example of a *RACI* chart

3.1.2 Process Identification Methods

Experiences show that understanding the receiver's intended usage of the process map is essential as it affects what data that should be collected and presented in the process map. Different roles require a different mix of information of *why-* and *how* type when working with a process map. For instance, a manager may need more *why-focused* information to better understand the organisation's purpose, but a worker may instead need more *how-focused* information to solve work tasks. Ljungberg has experienced that people working with process management tend to unconsciously have a practical focus on *how* things are done when conducting studies, if the purpose is not well thought-out beforehand [11].

Generally when humans analyse complex flows of information, a visual approach tends to improve their ability of eliciting similarities and relationship between processes [30]. This is an interesting fact as the process map therefore can be both used as a tool for analysis and to visualise an end result.

Work have been conducted to elicit communication processes for software teams with methods such as FLOW and MAPS introduced by Stapel et al. respectively Mellhorn et al. Both methods use a process oriented map whose semantics explained how communication is carried out. Both methods have with good result quickly created an initial map which served as a baseline for further investigation [39, 40].

3.2 Value Within the Software Industry

In order to use value-based process modelling for a software business is it crucial to define what value is in the software context. Based on the foregoing reasoning in *Section 2.1: Business Process Management* can everything that the customer appreciate be regarded as value.

It is not strange to further think that obviously the code in itself defines the core value. As the code base grows, is refined and is rewritten, must the value increase. This is not always the case though, as new code could interfere with the existing and cause new issues. An analogy with building a house is that after each section is completed the house will have a greater value. That is however not always true if the plans for the building are not followed and the house has to be rebuilt at a large cost.

Equivalently, in software Lauesen et al. conclude that it is frequent that projects fail or increase in cost due to that the wrong software is produced. This often comes as a result of insufficient requirement management and often the software has to be reworked, thereby driving up costs and causing delays. The cause for this is that customers rarely know exactly what to order and set requirement for [41]. A lot of successful software projects therefore involves consulting from experts that helps the customer to define its needs, which practically means creating and managing requirements.

The requirement space could also be considered as value as the lack of it would increase cost and decrease the quality of the software. This motivates that for example prototyping, where produced code is discarded in purpose of getting requirement insights, can be viewed as value producing.

To provide support in this activity the software industry has defined a set of generally applicable quality characteristics to strive for when defining the software's needs. They are summarised in the standard ISO/IEC 25010 [42] and consist of eight quality characteristic measures: *Functional Suitability*, *Reliability*, *Performance efficiency*, *Operability*, *Security*, *Compatibility*, *Maintainability*, and *Portability*. Processes delivering output that meets these goals should thus be regarded as *main processes* according to the definition of Rosing et al [9].

If requirements are considered to form a sort of value for software businesses is it natural to ask what makes up good requirements. Lausen recommends to validate the requirements space against ten characteristics based on IEEE 830-1998, see Table 3.2 [26, 43].

Processes whose output increases the quality of these characteristics should thus also be regarded as *main processes*. For example by reviewing the characteristics of Table 3.2 is it clear that activities like customer interviews and prototyping can be considered to be *main processes* as they elicit new requirements which contributes to completeness. In the same sense can the creation of an integration plan, which defines test cases, contribute to the verifiability of requirements.

Correct Each requirement reflects a customer need.
Complete All requirements necessary to describe the customer need are included.
Unambiguous All parties agree on the meaning of the requirement.
Consistent There are no internal conflicts between requirements.
Ranked for importance and stability Each requirement should state a priority and the expected frequency of changes.
Modifiable The requirements are easy to change while maintaining consistency.
Verifiable The requirement should be formulated so that it is possible to see whether it is met.
Traceable It should be possible to trace a requirement to the purpose it originates from, and where it is used in an implementation.
Understandable by customer and developer All parties should understand the content of the requirements.

Table 3.2: Good characteristics of requirements according to Lausen [26, 43].

3.3 Information Aspect of Value Focused Process Modelling

As a complement to streams of value have several authors introduced streams of information to process models [11, 44, 45]. Nash argues that information itself can provide insights about waste, hence indirectly helping organisations to focus on value delivery [44]. Ljungberg and Davenport view information as a supportive facilitator which is present throughout multiple mechanics like *main processes*, *control processes* and *support processes* [11, 45]. Ljungberg elaborates that information entities can be related to processes directly, but with the difference compared to value streams that they do not trigger processes to start [11].

In a recognised blog post Beams [46] experiences that *commit messages*, a piece of information that describes the changes for code in the version handling system *Git*, heavily influence how easy it is to re-establish the context of a section of code. As re-establishing is just a wasteful activity which should be reduced, this makes a perfect example of how information is facilitating a value producing process.

3.3.1 Communication

Communication seems to be used to influence and improve a wide range of areas within software development. Communication, which according to the definition by Clark in the work *Arenas of Language Use* [47], is a special kind of flow where information is transferred verbally or written between two people. Communication plays a central role in software development; case studies undertaken by Perry et al. indicate that developers spend around 15% of their day on communications [48]. A quasi experiment made by Dutoit et al. with 400 student in a development test-bed, which had characteristics in between an industry project and a controlled experiment, showed empirical evidence that communication metrics have direct influence on project outcome [49].

Communication as an exchange of information is a knowledge sharing activity between the two people acting as storage container for that information. This aspect is studied in the research field of *Knowledge Management* for software companies using agile practises. A literature review of 32 studies performed by Ouriques et al. showed that 81% of the practices for knowledge management use humans as storage, and communication as transfer technique for managing information [50]. Baskerville et al. confirms this by finding that organisations with teams spread out on separate sites experiences the challenges of communications to be among the most common ones [51]. The communication barrier is induced from a range of factors such as social and cultural differences, timezone mismatch, geographical split, etc. Relocating several sites to a central one is rarely an option as spread out sites gain additional competitive advantages such as competence, availability to customers, regional support, etc.

In the case of process mapping experience, Ljungberg and Lundgren [11, 35] have shown that the communication regarding the handover between processes is of great importance. By customising communication checklists covering the transaction between the roles active in the processes mistakes are greatly reduced.

Chapter 4
Research Method

Combitech needed a better overview of its core workflow to improve repeatability and management of iterative and sequential workflows. The problem was approached by creating a process map, presented in *Chapter 6*, which provided the needed overview for Combitech.

The process map was created by the *The ECPM Method* and *The ECPM Notation Language*, in short *ECPM* which stands for "Elias Christoffer Process Mapping", also designed in this study, and presented in *Chapter 5*.

The *ECPM Method* contains the activities used to collect, analyse and validate data. The *ECPM Notation Language* are the terms, design guidelines and graphical building blocks that are used to communicate the information harvested by the ECPM Method. The two parts can be used either together or on their own in conjunction with other frameworks.

The reason behind formalising a method on how to organise the work of creating a process map, separated from the research method, was the map's need of being updated to be useful. By developing both a process map and the tools to create and update it will Combitech be able to maintain a representative overview.

The research framework *Design Science Research* was used as a foundation to structure the study on. Design Science Research was suitable as Hevner et al. concluded that "the result from a design-science study in information systems is a purposeful IT artifact that addresses an organisational problem" [52].

The study was qualitative and had an explorative character, as the design of ECPM was lead by solving Combitech's problem. ECPM's design process and the work of creating the process map were integrated into each other, with drafts of the process map used to evaluate and improve ECPM iteratively.

The research was performed in three phases: "Preparation" (see Section 4.1), "Execution" (see Section 4.2), and "Evaluation" (see Section 4.3). Figure 4.1 illustrates the phases with their related activities and data flows.

The main purpose of the *preparation phase* was to make an initial design of the ECPM Method and Notation Language, which could be used to start the iterative development activity in the next phase.

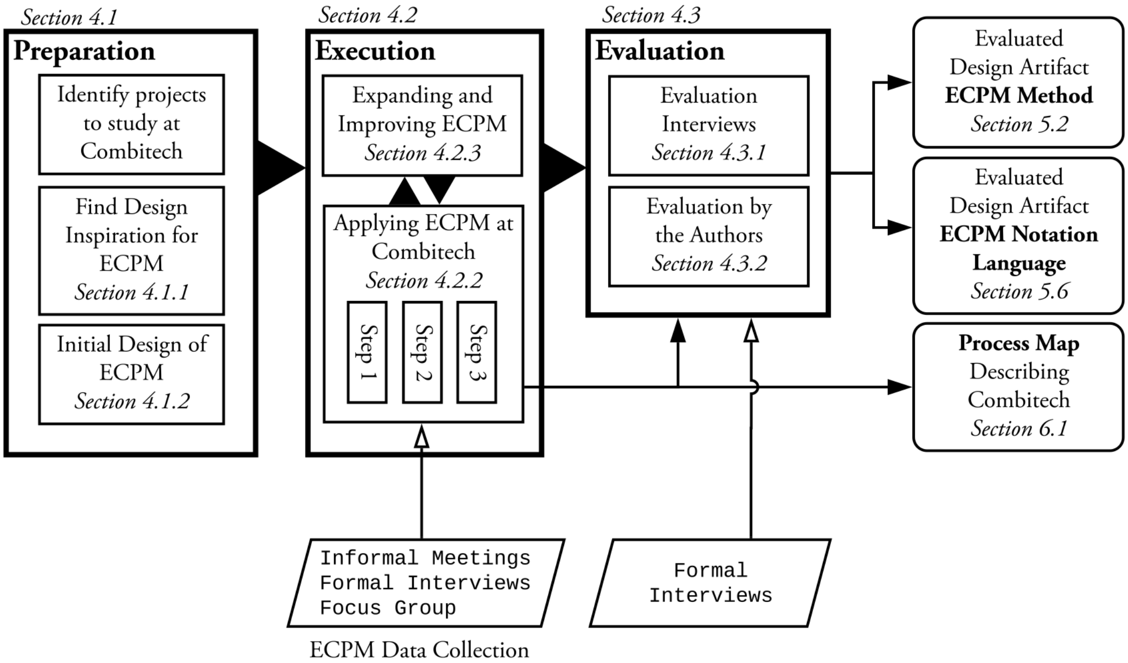


Figure 4.1: Overview of the research featuring the three phases "Preparation", "Execution" and "Evaluation", with their associated activities. The black triangle arrows represent the design flow, i.e. how *ECPM* was improved over time. At the far right are the final versions of the produced artifacts illustrated. At the bottom the data sources are illustrate to show how external data from the case study were used by the activities.

The *execution phase* produced both the ECPM Method and Notation Language, as well as the process map. The execution phase was made up by two activities; to *expand and improve* the ECPM Method and Notation Language, and to *apply* the ECPM Method and Notation Language at Combitech which created the process map. The experiences from applying ECPM drove the expansion and improvement of ECPM in an iterative fashion, until the execution phase had to be terminated due to time constraints.

The *evaluation phase* evaluated the designed artifacts ECPM Method and ECPM Notation Language, both directly and indirectly using the process map. The participants' experiences from the work with ECPM and the process map at Combitech were collected using evaluation interviews. The authors also evaluated ECPM, and their experiences were collected in this phase as well.

The knowledge base employed in the design process was mainly the fields of *business process management* and *software development processes*. Business process management brought the concept of value-creating processes with its associated properties. Software development processes contributed with procedures specific to the software field and management models such as the agile perspective. Other contributions came from the study of flows of information, specifically communication. The design of the notation language was built upon best-practices of graphical design.

4.1 Preparation of the Study

The purpose of the preparation phase was to make an initial design of ECPM, which could later be improved through iteration. When the preparation phase ended, ECPM had basic capabilities to collect and analyse data, and present it using a rudimentary notation.

4.1.1 Finding Design Inspiration for ECPM

Relevant literature was found through several different approaches:

- Recommendations of literature were received during supervision meetings at Lund University.
- As this thesis is interdisciplinary between software engineering and logistics, the authors took search terms specific to one of the fields, translated them into the other field's terms, and used the translated terms to search for corresponding concepts.
- The authors searched for articles within user experience and cognitive science to be able to design a notation that was effective in carrying data but easy to consume.
- The authors searched for best practices on how to conduct interviews, e.g. to avoid introducing bias.
- The authors searched for technical terms that was found at Combitech during the execution of the study to get the necessary industry knowledge.

In addition to these sources, the authors' different backgrounds from the educational programmes of Information and Communication Engineering respectively Industrial Engineering provided a wide base of relevant search terms. Their industrial experience from the telecommunications, IT-consulting, defence, and electronics sectors provided useful insights and experiences which in turn was used to search for relevant literature.

4.1.2 Initial Design of ECPM

Combitech had identified a project which had desirable properties and had been executed in an efficient way. Combitech wanted to replicate these characteristics in future projects. Also present at Combitech were two other similar projects which also had some of the desirable features.

Combitech's proposed solution was to create a process model based on a single project, and then improve it by including other projects. This process model would then be used to guide future projects towards success.

The authors decided, together with their supervisor at Combitech, to create this process model based on Project X (see Section 4.4), and then refine and validate it using interviews with other team members from the same and other projects (Project Y and Z) to make sure the map truthfully described these three projects. By initially creating a process model that covered the reference project (Project X), the scope of the explorative study was known and could be planned more easily.

The ideas above were formalised in the ECPM Method. The result was a three-stepped method which starts with the study of a single project, and ends with a validated process map based on several projects:

Step 1: **Create** a process map based on a reference project.
Step 2: **Refine** the map by including other projects.
Step 3: **Validate** the map using a focus group to reach consensus.

For each of these three steps the authors assigned activities for data collection, analysis, and validation. This set of activities was expanded as ECPM was executed to finally include all the methods described in *Chapter 5: Results: The ECPM Method and Notation Language*.

The initial design was complete just enough to enable the first data collection activities of the execution phase to be carried out, and hence something to let the iterative improvement process start from.

The initial design of the *ECPM Method* comprised an interview guide to be used for the interviews in the first step of the ECPM method. The interview guide was based on the interview guide included in the FLOW framework [34, 40], and expanded and optimised for this study. The interview guide was used for all of the interviews in the first step of ECPM, and then as a base for the upcoming interviews in the second step of ECPM. The guide can be found in *Appendix A: Interview Guide*.

The initial design of the *ECPM Notation Language* was inspired by the established VPM/IDEF0 models, described previously in *Section 3.1.1: Visualising and Interpreting Process Models*. The reason for this was that the VPM and IDEF0 were commonly used to solve the same base problem as the ECPM Notation Language intended to do.

4.2 Execution of the Study

During the execution phase the two activities *Applying ECPM at Combitech*, described in Section 4.2.2, and *Expanding and Improving ECPM*, described in Section 4.2.3, were performed as seen in Figure 4.1. These activities were ongoing in parallel during a period of three months at Combitech.

ECPM was applied to three projects at Combitech, namely “Project X”, “Project Y” and “Project Z”, resulting in a process map describing these projects. Further details on the projects are presented in Section 4.4.

The experiences and data derived doing so were injected into the *expand and improve ECPM* activity to fuel further development of the ECPM Method and Notation Language.

The execution phase was performed in an iterative fashion where both ECPM and the process map continuously evolved. An advantage of having an iterative design approach was that the study could be terminated at any time, while still providing useful results. This was a way of risk management, as the availability of the interviewees at Combitech could suddenly change as the participants would be assigned to client projects during the study. This risk was also managed by having each interviewee recommend who to interview next, instead of planning both the number of interviews and the participants ahead.

When the time limitation was reached, the process map and ECPM were set to form a baseline and were subsequently evaluated according to Section 4.3. The final version of the ECPM Method and Notation Language is found in *Chapter 5: Results: The ECPM Method and Notation Language*, the final version of the process map is found in *Chapter 6: Results: Applying ECPM at Combitech*, and the findings from the evaluation are found in *Chapter 7: Results: Evaluation of ECPM*.

4.2.1 Data Collection and Storage

The authors’ decision to work iteratively had certain implications on the data storage strategy. As the process map was used in the data collection and analysis activities in order to be refined, the map itself was the natural way of storing data. Audio files were used as an intermediate way of storing collected data from the interviews before the information was analysed and drawn onto the map.

As map drawing was one of the core activities in this thesis work, the authors needed powerful technology support in this area. The requirements on the drawing tool was that it should be priced within the budget of the thesis work, had support for the major operating systems, supported real-time collaboration, supported version control to track changes, could handle very large documents and was reasonably user-friendly. The tool that met these criteria was “Lucidchart”.

4.2.2 Applying ECPM at Combitech

By applying ECPM at Combitech a process model was created of the studied organisation. The process model was drawn using the ECPM Notation Language, and information was collected and analysed using the ECPM Method, both described in Chapter 5.

Practically, there were three types of drawing operations made during the process map creation:

- Drawing a new and independent illustration of a process.
- Adapting a previously independent drawing to fit logically with another existing illustration.
- Updating a previously drawn process to improve it according to newfound information.

For each addition to the map two versions were drawn independently by the researchers, from which the one considered to best use the ECPM Notation Language was chosen. This was done to make sure that the additions to the process map was quality assured.

4.2.3 Expanding and Improving ECPM

At the time when the project of applying ECPM at Combitech started not all activities in ECPM were defined. Likewise was not the notation language completed, as it just comprised an initial design inspired by the VPM/IDEF0 frameworks.

The focus of the expansion and improvement differed slightly between the method and the notation language. The development of the method was primarily driven by previous research and the experiences from applying the method at Combitech. The development of the notation language on the other hand was more data-driven with simulation and testing.

Addition of Activities to the ECPM Method

The ECPM Method consists of a set of activities, however not all of these were present at the start of the study. During the study activities were added and improved as need for them occurred. Two cases led to the development of new or improved activities:

I) The most common case was when data were discovered during the data collection which didn't have an appropriate analysis method. This case was therefore driven by the collected data and the case study.

II) The second case covered the opposite, when information to complete the process map were still missing, even after all currently existing analysis methods had been applied. By developing new analysis methods based on the knowledge base introduced earlier, new conclusions could be drawn from the existing collected data, i.e. the interviews.

Expansion of the ECPM Notation Language

The notation language included in the initial design of ECPM was not sufficient to illustrate all collected data, and hence the notation language also needed to be expanded during the execution. The methodology for expanding the ECPM Notation Language was divided into five steps:

i) The most common trigger for expansion of the notation language was a difficulty to illustrate something or a discovered ambiguity in the existing graphical elements.

Before proceeding with the expansion of the notation language was it important to clearly define the actual problem.

ii) Based on the defined problem, literature was consulted to find existing solutions or at least ideas for a solution. Thoughts and insights from academia as well as the industry were considered.

iii) The design of the entire notation language, thereby including all additions, was optimised for describing *why* something was done in favour for *how* something was done. Examples of this are the naming of processes, the level of detail the map was drawn on, the exclusions of checklists, etc.

The proposed addition to the notation language was also checked to not cause an implementation that was too physically large and complex to draw and understand. Particularly the illustration of logic and information flows were checked to not produce too much clutter.

iv) The addition was tested against the problem definition but also simulated on fictive or similar use cases to make sure it solved the intended problem.

Extra complex parts of the notation were tested together with participants from Combitech before it was included in the notation.

v) If the addition was proved to be useful by passing the tests was it included and the notation language expanded.

4.3 Evaluation of ECPM

When the project of expanding and applying ECPM at Combitech was terminated, and the final version of the process map had been presented for the team, the evaluation phase of the study followed, as illustrated in Figure 4.1. The purpose was to investigate how well ECPM performed, primarily by studying the results which ECPM had produced, namely the process map.

The evaluation carried out was inspired by *Design Science Research* by Hevner et al. and were conducted with two distinct directions: evaluation interviews, described in Section 4.3.1 and evaluation by the authors as explained in Section 4.3.2.

4.3.1 Evaluation Interviews

The performance of the ECPM Method and Notation Language was studied by a series of evaluation interviews with the participants of the study. An assumption was made that if the interviewees found the process map to be of good quality and useful, it would be possible to assume that the same applied to the method and notation language which had *created* the process map. The evaluation interviews therefore circled around the process map.

The interviews contained thirteen questions, found in *Appendix B: Evaluation Interview Guide*. The interviews were performed individually with each team member. A printed process map was accessible during the interviews in order to improve the accuracy of the answers.

4.3.2 Evaluation by the Authors

The authors carried out an evaluation which had seven different perspectives, explained below.

To ensure as many quality aspects as possible were covered in the evaluation of the artifacts, ISO 9126, describing quality characteristics for software-related artifacts, was used as guidance during this evaluation. The standard divides quality characteristics into six major fields: functionality, reliability, usability, efficiency, maintainability, portability.

There have been critics questioning the reliability of this rather old standard, especially when deriving *quantitative* measurements from it [53, 54], but in this particular case it is used in a *qualitative* context and not directly as a measurement, so it can still be regarded as relevant.

The ECPM Method

The ECPM Method was evaluated as follows:

- The ECPM Method has an inevitable built-in structural bias as the process mapping needs of Combitech propelled the development of the method. It is important to have awareness of this bias and how it affects the characteristics and focus of the method, as ECPM was designed to be suitable for use in other mapping projects at other organisations. The authors tried to identify the bias of the ECPM Method by studying the static structure of ECPM.

  For instance could the number of activities that concerns for instance objects be compared with the number of activities that concerns for instance roles. The relation between these numbers could then be compared to find the focus of ECPM.

- During the study metrics of time usage were collected in order to create conclusions about the performance of different parts of ECPM. This was made through analysis of a simple logbook which kept information on what had been in focus on each week. What made this method hard was the fact that a lot of analysis activities were integrated into each other.

- Scalability was studied with the following two aspects in mind: "How fast will the complexity increase if more *projects* are included in the study?" and "To what degree is it possible to add more *participants* to the study?".

  While applying ECPM at Combitech the activities in ECPM were continuously reviewed on how well these allowed ECPM to be scaled. The results were summarised and concluded in the end of the study.

- The authors also collected experiences while using the ECPM Method that covered a wide range of topics. The experiences were foremost issue related, such as what ECPM could not do or what did not work well. The experiences were collected as notes and finally compiled at the end of the study.

The ECPM Notation Language

The ECPM Notation Language was evaluated as follows:

- The final process map, drawn using the final version of the notation, was examined to see what real-world usage of the notation language could look like. It was checked if the purposes of each component, element and rule from the notation were used as intended. The evaluation were both quantitative, i.e. by counting the number of objects in the process map, and qualitative, i.e. checking if a certain logic had been used and if so to what extent.

- There are a lot of previously existing models and concept which addresses similar topics as the ECPM Notation Language does. By comparing these designs to the final version of the notation language, strengths and weaknesses compared to existing models could be identified.

- The authors' experiences from the usage of the ECPM Notation Language in the work of developing the process model at Combitech were collected as the project was ongoing, and the most important were concluded when the study ended.

4.4 Case Data

The problem identified by Combitech, described in *Section 1.2: Case Company: Combitech AB*, was studied via six employees that together were members of three projects. These persons participated in informal meetings, formal interviews, and a focus group.

The three projects that were studied are due to the need of confidentiality referred to as "Project X", "Project Y", and "Project Z". These projects had been running for various length of time. Characteristics for Project X, Y and Z are found in Table 4.2, 4.3 and 4.4 respectively. The projects were chosen as they represented Combitech's workflow well, and had team members available for data collection activities at the office. All three projects were decided on before the study started.

The six employees that participated in the study are referred to as "Person A" to "Person F", and are listed in Table 4.5. In Table 4.1 an overview of the data collection activities is shown. The interviewees were chosen as the study went along by letting each interviewee recommend who the next interviewee would be.

Table 4.1: Overview of the data collection activities.

ECPM Step	Data Collection Activity	Person Involved	Project Represented
Create	Interview 1	D	X
Create	Interview 2	C	X
Create	Interview 3	A	X
Refine	Interview 4	A, B	X
Refine	Interview 5	B	Y
Refine	Interview 6	A	Z
Refine	Informal Meetings	A, F	X
Validate	Focus Group	A, B, E, F	X, Y, Z

Table 4.2: Project X Characteristics

Type of project:	SDaaS
Domain:	Medical Equipment
Number of people in project:	5
Number of active developers:	3
Distributed:	No
Main source of requirements:	External and Internal
Software Development Process:	Scrum
Duration of project:	5+ years
Keywords:	Explorative, Global Usage, Cloud Application

Table 4.3: Project Y Characteristics

Type of project:	SDaaS
Domain:	Manufacturing
Number of people in project:	4
Number of active developers:	3
Distributed:	No
Main source of requirements:	External
Software Development Process:	Scrum
Duration of project:	3 years
Keywords:	PLC, Windows, UX

Table 4.4: Project Z Characteristics

Type of project:	SDaaS
Domain:	Automotive
Number of people in project:	9
Number of active developers:	5
Distributed:	Yes
Main source of requirements:	External
Software Development Process:	Scrum
Duration of project:	1 year
Keywords:	UX

Table 4.5: Interviewees at Combitech

Name	Role	Years at Combitech	Years in Industry
Person A	Scrum Master, Developer	3	20
Person B	Developer	3	9
Person C	Architect, Developer	3	11
Person D	Product Owner	12	24
Person E	Scrum Master	0.5	11
Person F	Department Manager	2	20

Chapter 5

Results: The ECPM Method and Notation Language

This chapter presents the "ECPM Method and Notation Language", in short "ECPM". ECPM is an abbreviation of "Elias Christoffer Process Mapping". ECPM consists of two parts: a method for collecting and assessing information, introduced in Section 5.2 and further explained in Sections 5.3 to 5.5, and a notation language which brings a graphical toolset that enables the creation of process maps using the information compiled by the method, described in detail in Section 5.6. A list of terms and definitions used in ECPM can be found in Section 5.1.

Scope of Applicability

ECPM was designed to be used at a consulting company that provides software development as a service. When ECPM is applied to an organisation it produces a descriptive or "As-Is" value-based process map with a holistic viewpoint over the studied projects. ECPM does not in itself improve an organisation; it creates a model which later can be used by management, for instance for organisational improvement. If the model is used to describe projects which the organisation want to replicate in the future could ECPM be used prescriptively and illustrate the "To-Be"-state of the company.

The value perspective means that all processes which create or increase the value of anything that the customer is willing to pay for are illustrated. The value perspective is generic, so it is not limiting the applicability of ECPM to be used only within software. For more details of what value means specifically for software, please refer to *Chapter 2: Background* and *Section 3.2: Value Within the Software Industry*.

5.1 ECPM Terms & Definitions

ECPM	An abbrevation of "Elias Christoffer Process Mapping". "ECPM" is used to refer to both the "ECPM Method" and the "ECPM Notation Language".
ECPM Method	The collection of activities that collect, analyse and validate data.
ECPM Notation Language	The collection of graphical building blocks used to visualise the findings from the ECPM Method.
Process	A generic superclass, never used in the notation.
Main Process	Processes which produce value, in line with Rosing's definition [9].
Control Process	Processes controlling main process, in line with Rosing's definition.
Support Process	Processes enabling main process, in line with Rosing's definition.
Core Process	A single top level process describing a clearly separated part of the value chain.
The Process Map	The entire map including all associated processes, descriptions, explanations and legends.
Core Process Map	The top level map covering all core processes, control processes and support processes.
Process owner	Position which is ultimately responsible for the performance of the process. Manages and keeps the process map updated and relevant.
Object	An entity produced by a process, represents a value.
Control Object	An entity used as an enabler to a control or decision technique.
Legal Object	An object that can cause legal issues if not handled correctly.
RAM	"Responsibility Assignment Matrix", a matrix displaying how responsibilities for processes are divided between roles.

Perspectives of a Project

To improve interpretation and understanding of the process maps produced, in addition to the terms and definitions introduced earlier, a conceptual model stitching the three perspectives *Value creation*, *Project Timeline* and *Realisation* together is shown in Figure 5.1.

The *Value creation* perspective shows *what* value that is demanded by the customer, i.e. what should be delivered, and the dependencies active when doing so. This is what the *main processes* displays on the process maps produced by ECPM.

The *Project Timeline* perspective shows *when* a certain value is created during a project's lifetime, i.e. in which scope of time certain processes are executed, and to what extent during a specific project. As time goes by and projects become more mature, a more refined value is likely produced which can be seen in the upper graph in Figure 5.1 as the intensity of Core Process B and C is higher than for A towards the end of the project.

Finally, there is a perspective of *Realisation* answering *how* and by which means value is produced, e.g. the decision making, planning and follow-up needed for execution.

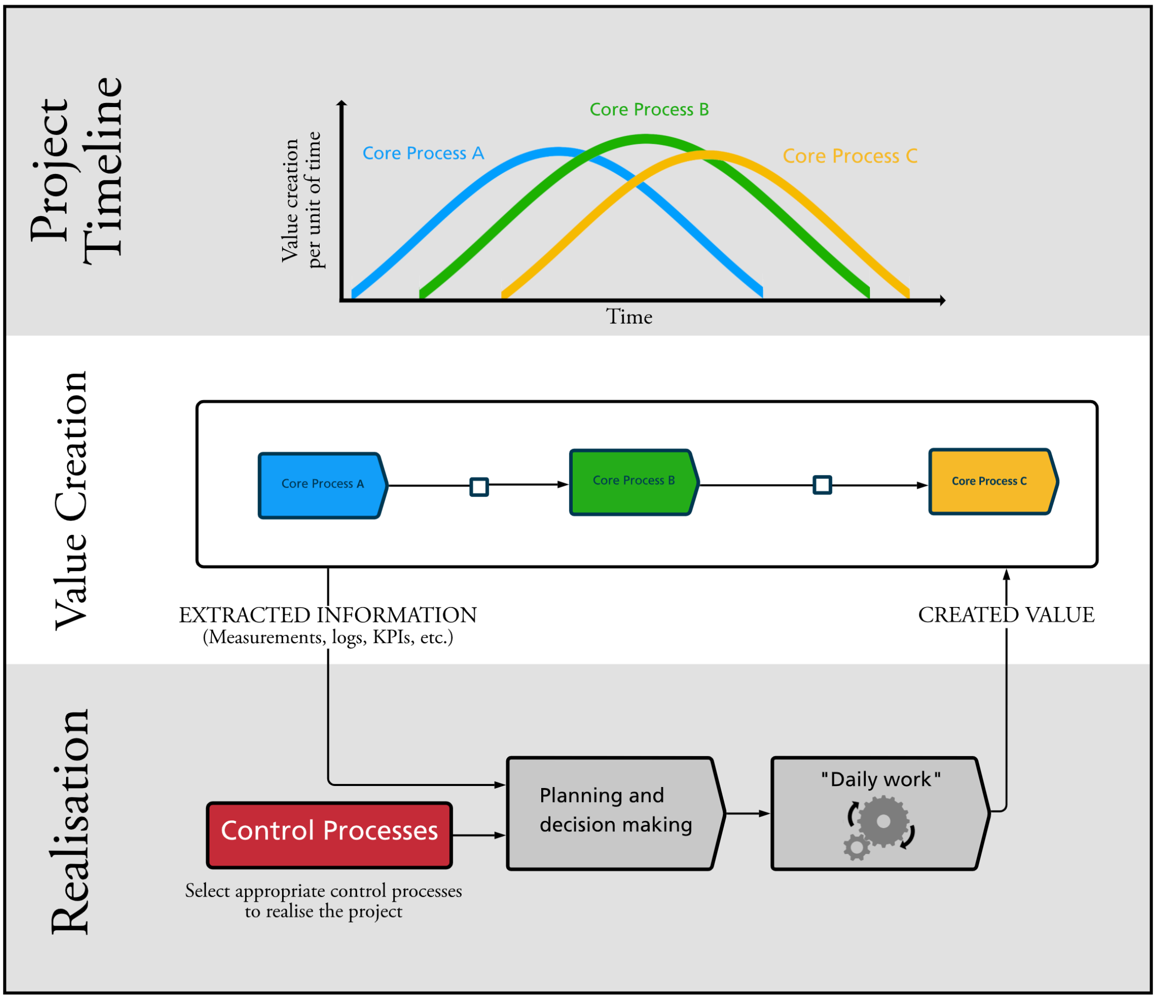


Figure 5.1: The figure demonstrates how the value perspective relates to a project's timeline as well as the realisation, i.e. how activities are executed and monitored.

5.2 ECPM Method

The ECPM method consists of three distinct steps: *create*, *refine* and *validate* with associated activities, all summarised and illustrated in Figure 5.2. The background for this design decision is described in *Section 4.1.2: Initial Design of ECPM*.

The first step *create* aims to produce an initial process map based on a reference project carried out by the studied team. In the second step *refine* the initial map from the first step is broadened to cover more projects. The third and final step *validate* is meant to review the produced process map to ensure quality.

The activities included in all steps are used in an iterative manner at the researchers' discretion until each step's goal is reached. Depending on the studied organisation's size and the scope and ambition of the process mapping different number of iterations is needed.

A reference project is characterised by that it reflects either how the organisation is currently working, or how it would like to work in the future. The aim is to be able to repeat the success once delivered by the reference project in all future projects. Projects similar to the reference project should also be identified. These projects does not need to have all the characteristics of a reference project, but be similar enough to help expand the process map to be as generic as possible.

The possible persons to interview needs to be identified, to make sure that the selected projects are possible to include in the mapping study. If the projects doesn't have any possible persons to interview, the projects can not be used as the method is built upon interviews. The researchers specifically needs to verify that they will be able to book interviews with the interviewees, and not be restricted by for example security and secrecy measures. It may be useful to plan the study with a representative from the organisation who have good insight in such issues and authority to make such decisions.

The project of applying ECPM should further be prepared by setting delimitations of the process map. Three questions are suitable for deciding this:

1. Where should the process map start and end?
2. How detailed should the map be – should it just be a core process map, or contain several levels, with the lowest level comprising a detailed checklist?
3. What processes should be included – for instance, should support processes be included?

A suggestion is that the researcher should strive to quickly make a preliminary analysis so that information can be stored directly in the map artifact. Intermediate data storage formats such as audio recordings, photos of sketches on whiteboards and text notes are probable to be used. By swiftly processing the data into a process map it is easier to decide if more data collection activities are needed, as all participants can see what information that have been found.

All activities in ECPM, described in the upcoming sections, have an introduction which describes what problem the activity solves, what the solution is, and what data the activity outputs. The purpose of these introductions are to facilitate for the researcher to select the most suitable activity for the problem he or she is currently facing.

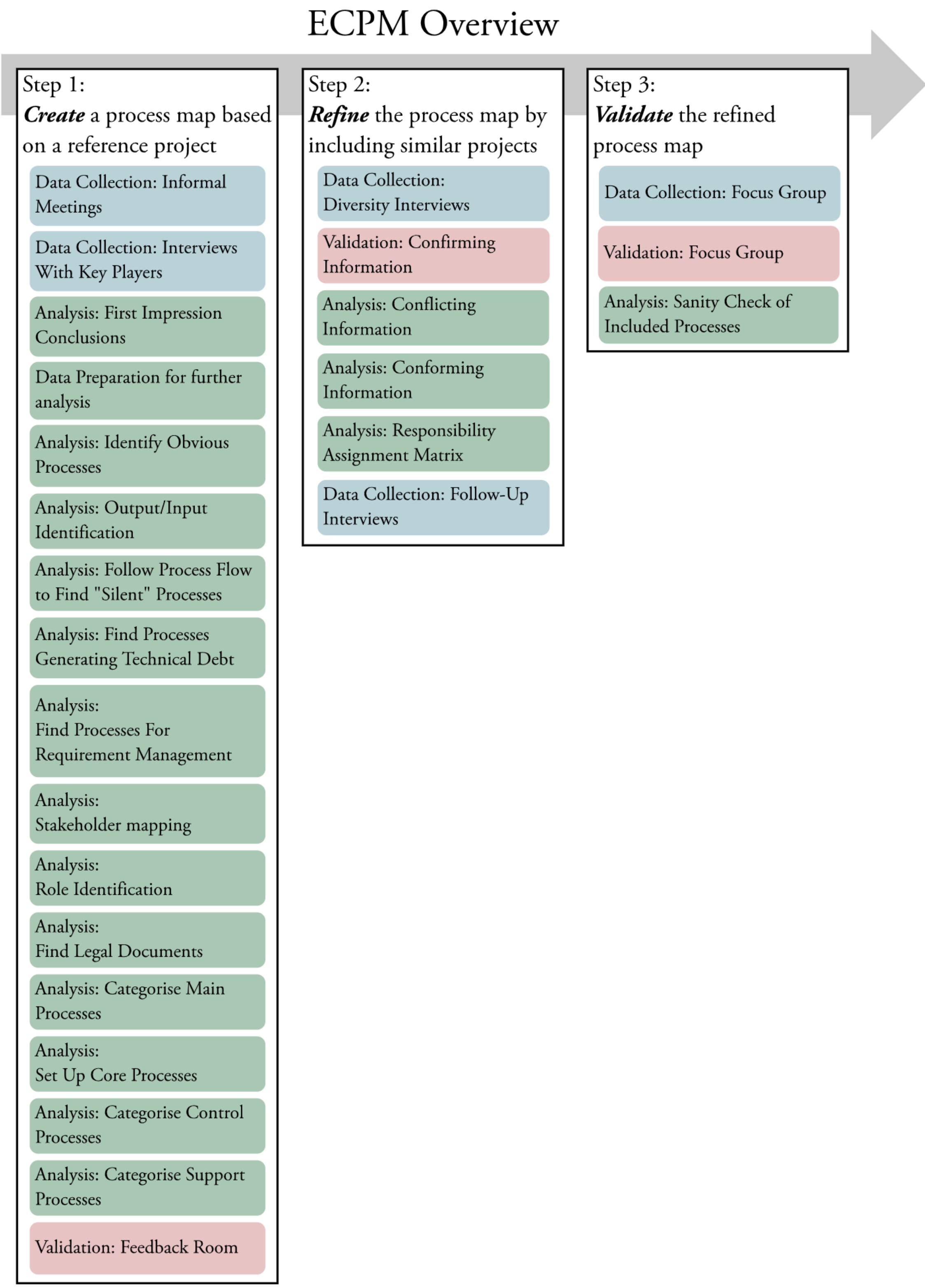


Figure 5.2: Overview of the three steps in ECPM. These three steps are in turn made up of activities which either are of "Data Collection", "Analysis" or "Validation" type.

5.3 ECPM Method Step 1 of 3: Create a Process Map Based on a Reference Project

The goal of this step is to produce a draft of the entire process map of the studied organisation. The purpose for the draft of the process map is to efficiently provide an initial overview of the organisation and thereby establishing the scope of the upcoming process mapping activities. The map is then used as a tool for organising the future findings. This enables the researchers to refine the map by subsequently adding collected data and facilitate analysis.

A reference project, which represents the state of the organisation, shall be chosen as the initial information base. This reference project could preferably be a previous well-executed project which the organisation would like to reproduce.

The authors consider that awareness of the entire value creation chain is crucial to counteract the *process paradox* accounted for in *Section 2.1.1: Common Challenges with Process Management*.

If information of conflicting nature is collected during the mapping of the reference project, and not possible to resolve directly, the conflict should be stored to later be resolved in the activity *Section 5.3.17: Feedback Room*, at the end of step 1. Conflicts are suitably stored on the process map as different variants drawn using the Notation Language.

5.3.1 Data Collection: Informal Meetings

Problem:	By only visiting the studied organisation to conduct formal interviews is it possible that contextual insights and spontaneous thoughts are missed in the data collection.
Solution:	The researchers should be present at the studied organisation whenever possible.
Output:	Unstructured notes

Procedure

"Informal meetings" is an effective tool that complements the formal interviews. As the purpose of the entire mapping method is to create an overview of how the organisation works, the more information that is collected, the better.

For instance, if the interviewee wants to add facts which doesn't fit into the formal interview questions, can informal meetings be used to collect such information afterwards without disturbing the formal interview.

This activity thereby lowers the requirements on the formal interview guide's coverage, i.e. it doesn't have to be perfect. The staff within the studied organisation may sometimes also take initiative for meetings with the researchers if they have opportunities for it. One way of improving the odds of such initiatives is to create a "gallery" in which the latest version of the process map is put on display for public view at the office. As this activity is not structured will the data produced cover a wide range of topics.

5.3.2 Data Collection: Interviews with Key Players

Problem:	Exhaustive data collection from the reference project is needed.
Solution:	Interview members of the studied organisation
Output:	Audio files

Procedure

This activity is about interviewing team members from the reference project in a structured way. The first interviewee in the formal interview series should ideally be someone with a complete overview of the entire project, for instance a project manager. Ideally has this key player been recommended and actively selected beforehand based on contextual insights gained during a series of *informal meetings*.

Each interviewee should recommend who to interview next in the team. The researcher still needs to keep in mind that people with various roles in the organisation needs to be interviewed, and just take the recommendations as inspiration.

Interview Guide

Before a series of formal interview is carried out an interview guide should be written. The interview should follow a semi-structured pattern with the written interview guide as its base. When the interview guide is written it should be tested and the time needed to follow it through estimated to make sure it is appropriate in length.

The interview guide is primarily used in advance to think through the interviews to "guarantee" full completeness. During the interviews the entire interview guide is impractical to use, as it is to detailed. Instead it is better to use a checklist to make sure that all questions are answered and full completeness is achieved.

In *Appendix A: Interview Guide* an example, the one used at the interviews at Combitech, is found.

Interview Technique

The interviewer needs to be prepared for conducting interviews. Apart from being familiar with the interview guide with its questions, contextual and industrial knowledge are necessary to understand what the interviewee is accounting for. It is also recommended that the interviewer studies relevant literature on general interview techniques, for instance "Qualitative Research Methods: A Data Collector's Field Guide" by Mack and Woodsong [55], and "Qualitative interviewing: The art of hearing data" by Herbert J. Rubin et al. [56].

Recording

In order to maximise the effectiveness and efficiency of the activity, the interview should be recorded. Make make sure to get necessary permissions to record, and plan for safe storage of the audio files as they may contain sensitive information.

5.3.3 Analysis: First Impression Conclusions

Problem:	The interviewers' thoughts and conclusions drawn during and directly after the data collection activities are volatile and easily forgotten
Solution:	Quickly write down these thoughts afterwards
Output:	A paper with notes and illustrations

Procedure

All the information that the interviewee provides is not captured in the audio recording, for instance body language. After each interview or other data collection activity should the interviewers quickly gather and summarise the first impressions together. These impressions includes thoughts and conclusions drawn during the interview, associations, etc that may easily be forgotten. It is also possible to prepare short discussion questions beforehand or use the interview guide to sort and relate this information to.

This kind of analysis activity have been shown good result as long it is performed in close relation to the sessions [57]. It is also useful to have written notes from the interview as a backup to the recording.

5.3.4 Preparation of Data for Further Analysis

Problem:	It is difficult to analyse the data collected during the interview when it is stored in an audio file.
Solution:	Transcribe the audio recording of the interview to text.
Output:	A spreadsheet with time-stamps and notes

Procedure

The next activity after a data collection activity is done and initial impressions concluded is to prepare the collected data, i.e. the recording and the first impression conclusions, for analysis.

Quick Log

There are several ways to transcribe interview recordings. The authors suggest that at first a “log”, not a full transcription, is written. The authors recommend that the log is created in a spreadsheet, which may use the interview guide's questions as its base. The spreadsheet may have columns for the interviewee's answers, the elapsed time in the audio file when the questions are answered, and tags to categorise what kind of information is discussed at a certain time. The tags makes it easy to find information concerning a specific topic.

The logs are then used to structure, find and extract all interesting information from the audio file.

One could argue that there is a risk that the log excludes vital information, but that is not a problem as the recordings will be listened to many more times, with the log as assistance.

The estimated time needed to write a log as described above is about 1:1, e.g. 1 hour of recorded content takes approximately 1 hour to write a log for.

In-depth Listening

After the quick log activity is completed is it suggested to further prepare the data for analysis by expanding the previously created log file. The authors suggest that this work is done with all researchers participating in order to find as much information as possible.

More columns can be added in the spreadsheet, for instance follow-up questions for clarification, conclusions, identified processes, role descriptions, etc.

The recordings from the interviews are still used as source material, with the created logs as assistance to sort information and to navigate in the audio file efficiently.

5.3.5 Analysis: Identify Obvious Processes

Problem: There is a need to elicit processes.
Solution: Deduce well-established processes with their respective input and output objects.
Output: Updated Process Map with new processes that have unambiguous descriptions.

Procedure

This activity is about finding the obvious processes and activities mentioned in the interview logs. Sometimes a processes can be talked about without have been given an explicit name by the interviewee. In these cases it is needed to make up a temporary process name.

This analysis is important as the output serves as a foundation to relate and compare other information to.

5.3.6 Analysis: Output/Input Identification

Problem: Objects can have been identified in data collection and analysis activities, but might not have been related to a process.
Solution: Elicit new processes which produces or uses the detached objects.
Output: Updated Process Map with new processes.

Procedure

If the researcher somehow knows that a certain object is being produced, he or she can safely assume that there is a process that creates that object. The researcher can now ask him or herself what is needed to create this object, and a new process is found. Be aware to not to mix up *objects* with *control objects*.

5.3.7 Analysis: Follow the Process Flow to Find "Silent" Processes

Problem:	Contradictory output and input objects exists between processes
Solution:	Elicit new processes that introduces new object flows.
Output:	Updated Process Map with new processes and new flows between existing processes.

Procedure

When eliciting processes is it natural that some are more obvious and visible than others. An example regarding cooking is that it is easy to "see" the process of frying, but the process of moving deep-frozen foods from the freezer to the fridge the night before might not be that obvious.

One method to find such processes is to look at the start and end of the process model, and making sure that the input and output in between the processes are logically correct. In the case of cooking, we might see a "served dinner" in one end, and can then move through the process to find that we need ingredients, and ask ourselves where they come from, and there find the process to move the foods from the freezer.

This analysis will partly be based on assumptions as the introduced processes glue the map together. Make sure to verify everything introduced as soon as possible to counteract the risk of basing further assumptions on weak ground.

5.3.8 Analysis: Processes Generating Technical Debt

Problem:	Technical debt, which also can be considered to be "value debt", needs to be visualised.
Solution:	Analyse and examine all processes which generate or resolve technical debt.
Output:	Updated Process Map with symbols that illustrate which processes that generate or resolve technical debt, and where information about the technical debt is stored.

Procedure

The concept of *Technical Dept* (TD) is explained by Zengyang et al. [58] who made a mapping study of the field:

> "Technical Dept is a metaphor reflecting technical compromises that can yield short-term benefit but may hurt the long-term health of a software system."

Technical debt can be regarded as a kind of loan of value which is invested but needs to be paid back later on. In the context of the process map the technical debt, i.e. "loan of value", should be visualised in correlation to the processes which either create technical debt or resolve it.

The work of Zengyang et al. have introduced ten categories which is used to identify and mark technical debt on the process map:

Requirements TD refers to the distance between the optimal requirements specification and the actual system implementation, under domain assumptions and constraints.

Architectural TD is caused by architecture decisions that make compromises in some internal quality aspects, such as maintainability.

Design TD refers to technical shortcuts that are taken in detailed design.

Code TD is the poorly written code that violates best coding practices or coding rules. Examples include code duplication and over-complex code.

Test TD refers to shortcuts taken in testing. An example is lack of tests (e.g., unit tests, integration tests, and acceptance tests).

Build TD refers to flaws in a software system, in its build system, or in its build process that make the build overly complex and difficult.

Documentation TD refers to insufficient, incomplete, or outdated documentation in any aspect of software development. Examples include out-of-date architecture documentation and lack of code comments.

Infrastructure TD refers to a sub-optimal configuration of development-related processes, technologies, supporting tools, etc. Such sub-optimal configuration negatively affects the team's ability to produce a quality product.

Versioning TD refers to the problems in source code versioning, such as unnecessary code forks.

Defect TD refers to defects, bugs, or failures found in software systems.

It is not within the scope of ECPM to answer how to mitigate and solve TD. Please refer to the work of Zengyang and others to seek guidance in this subject.

5.3.9 Analysis: Find Processes for Requirement Management

Problem: The studied organisation aren't aware of all its requirement management processes.
Solution: Examine which processes that covers basic activities related to requirement management.
Output: Updated Process Map with new processes that especially focuses on requirement management.

Procedure

The primary goal with this activity is to elicit new processes that concerns requirement management, and the secondary is to identify which of the previously found main processes that generates requirements. This is done by screening the process map for complete software components, and backtracking in the value chain to find the sources of their requirements.

The assumption on which this activity is built upon is that in order for a software product to have requirements with characteristics according to IEEE 830, introduced in *Section 3.2: Value within the Software Industry*, must there be processes that somehow generates requirements with these characteristics. By looking at the characteristics presented in the aforementioned section, and assuming that the software components have requirements that meets these categories, we can start to look for the missing processes that have provided the software component with requirements of this category.

For example, ask yourself: "Where does the requirements concerning *completeness* of this object come from?"

The form of the processes that produces requirements will probably vary from formal reviews to intuitive decisions based on individual employee's experience.

5.3.10 Analysis: Stakeholder Mapping

Problem: Team members' communication and influence patterns are strongly related to what they work with and are therefore a possible source to elicit processes and other information from.
Solution: Perform a stakeholder analysis which reveals connections internally as well as to external organisations.
Output: The "Stakeholder Mapping"-chart used while analysing.

Procedure

The "stakeholder mapping" method, originally formalised by Mendelow [59], maps the relation of interest and power to stakeholders from the interviewee's viewpoint in a two-dimensional chart. The chart is useful both during the interview to structure data , as well as to improve understanding of the organisation during the analysis.

More precisely could the thinking in the dimensions of influence and information be used to elicit unknown roles, finding out where the same role is referred to by different names and indicate who exercise control over the interviewee in the organisation.

5.3.11 Analysis: Role Identification

Problem: Not all roles are defined on the process map yet.
Solution: Inventory of roles in the organistaion and their associated processes.
Output: Updated Process Map with roles added to every process.

Procedure

To make the process map useful and to be able to manage the organisation is it essential to identify which roles that exists in the organisation. The roles should be separated from the persons in order to simplify staffing and lower the cost when on-boarding new members to the organisation. When eliciting roles the following questions is good to ask:

When are roles added to a project?

Does a project members staff several roles?

If a project member would be removed from a project, what processes will not run properly?

If the organisation is small there might be more roles, e.g. developer, tester, project manager, than employees. If the organisation is large there might either be more employees than roles, or more detailed and specialised role descriptions.

It is important to decide on the correct role names. The same role may be referred to by different names by different individuals. The wrong terminology and definitions may also be used in the organisation, for instance if the organisation uses a framework with very specific role definitions, but not uses the correct role names.

5.3.12 Analysis: Find Legal Documents

Problem: Some processes or objects may have strict legal obligations that the organisation must be aware of.
Solution: Examine all objects to find those with business or legal focus.
Output: Legal documents are specially marked on the Process Map.

Procedure

Screen all objects on the process map and analyse if they contain any aspects which can impose legal obligations for the organisation. If that is the case, mark them specifically with the *legal object* symbol. Typical legal objects related to software are business agreements, service level agreements, time estimates and data protection legislation.

5.3.13 Analysis: Categorise Main Processes

Problem: Processes which contributes to the value creation need to be clearly separated from other processes.
Solution: Backtrace concrete output to find processes that provides value to the client.
Output: Updated Process Map with Main Processes

Procedure

Analyse all the processes elicited and identified during earlier activities individually to find out if they produce value for the customer. If that is the case, categorise the process as a main processes and keep it in the process map.

What to classify as value can be vague in some situations. If the process needs to be included to make sense of other main processes, hence providing improved understanding and overview of the whole process map, should it probably be considered to be a main process. Another case to consider is when the process or related output plays a central role in the realisation. An example can be a configured toolset everyone are using, for instance a continuous integration platform. It is hard to directly connect the integration platform directly to the external customer, but omitting the processes managing it will make it hard to use the process map in everyday work.

5.3.14 Analysis: Set Up Core Processes

Problem: It is hard to get a clear overview of the organisation without arranging processes in levels.
Solution: Condense the main processes to a bare minimum which fit the core process criteria.
Output: Updated Process Map with Core Processes

Procedure

Take all identified main processes and divide them into groups based on what value they produce. These groups form a set of core processes. Three to five core processes will describe most organisations.

Every core processes shall thereafter be given individual short descriptions stating what purposes that brings these grouped main processes together (an example is found in Figure 5.3).

When all descriptions are complete, perform a comprehensive review of the distinctness between the core processes; it is important that the core processes convey the business or value edge of the organisation.

This review will probably reveal the need for moving main processes between the core processes to ensure consistency, or the need for splitting up the core processes even more and create new ones.

Creation and editing of core processes is somewhat of an iterative activity that ends when the core processes can give a good overview of the studied organisation.

5.3.15 Analysis: Categorise Control Processes

Problem: Some processes do not create value for the customer, but are necessary to manage the company and could be regarded as control processes.
Solution: Search for processes that influences how other processes are performed.
Output: Updated Process Map with Control Processes clearly separated from Main Processes

Procedure

All processes and activities that manage and decide how value is produced by other processes are considered to be *control processes*.

A simple way of deciding whether a process is a control process is to ask oneself *"Can I swap or remove this process but still produce the same value to the customer?"*. If this holds true is the process probably a control process.

Different control process have different focus; from strategic level to operative. This is needed to take into consideration when trying to identify them. The level of the elicited control processes will be dependent on what roles the interviewees has; for instance a developer might not see economical control processes, but maybe data storage legislation.

To what extent the control processes should be described varies between different organisations' needs. The level of detail of control processes printed out on the map can be varied. Either can the control processes be briefly mentioned on the Core Process Map, or they can be deeply integrated all the way through the map to the lowest level.

5.3.16 Analysis: Categorise Support Processes

Problem: A set of processes are critical for the organisation's performance, but are not directly a part of the value stream neither are they control processes.
Solution: Search for processes which supports the execution of other processes.
Output: Updated Process Map with Support Processes clearly separated from other processes.

Procedure

The categorising of support processes is performed after that the main and control processes have been identified. Processes not yet categorised are either regarded as support processes or discarded. Support processes can also be used to enrich the process map to cover important aspect not taken care of so by main and control processes.

When support processes are identified they are incorporated on to the Core Process Map as described in Figure 5.8.

5.3.17 Validation: Feedback Room

Problem: The process map must reflect the reference project. The map, in its current shape, might not do so.
Solution: Verify collected information with new data sources within the reference project.
Output: Audio files and notes on the process map.

Procedure

When the map is considered to be finished for the scope of the first iterations of Step 1, i.e. the drawing based on a single reference project, the map should be printed and validated.

The validation of Step 1 should be done by presenting the map to not previously interviewed members of the same project that have been studied in the interviews. Optimally, find the same roles as those interviewed earlier, for instance a tester, a developer, a project manager, etc, in order to get a validation of all perspectives.

Let the participants of the validation check if they can find and follow the value creation flow for their role on the map. It is important that they do this from the perspective of the specific project, and not from another project they are working on.

This activity can be considered to be a small focus group. Focus groups validates and naturally finds conflicting information if the participants have roles diversified enough. [60]

When there are a need for changes and improvements on the map, add notes directly on the map and make a new iteration of the map when the session is done. The session should be recorded.

5.4 ECPM Method Step 2 of 3: Refine the Process Map by Including Complementary Projects

The goal of Step 2 is to refine the process map drawn in Step 1 by including other projects in addition to the reference project. Efforts shall be concentrated on adding data from mutually exclusive activities and positions not present in Step 1, and improving the map where it is incomplete and lacks depth. The purpose is solely to produce an as complete process map as possible, which could act as release candidate to be validated in Step 3.

5.4.1 Data collection: Diversity Interviews

Problem:	More data needs to be collected in order to make the draft of the process map more precise.
Solution:	Perform additional interviews.
Output:	Audio files, notes on the process map

Procedure

Step 2 of ECPM also contains interviews. These interviews differ from those in Step 1 in that they are more based on the process map instead of an interview guide to directly target the aforementioned shortcomings.

These interviews starts with the gathering of general characteristics of the new project, and is then continued with letting the interviewee look at the process map which is currently based on the reference project, in order to find diversities.

Ask what part of the interviewee's project does and does not fit to the process map, and maintain focus on differences. If it is possible to conduct several interviews concerning the same project is it suitable to ask another interviewee about the conflicting information found in a previous interview.

5.4.2 Validation: Confirming Information

Problem:	Which data that are valid between the different studied projects needs to be confirmed.
Solution:	Identify statements from interviews which directly or indirectly prove confirmation.
Output:	Some processes are explicitly marked as confirmed on the process map.

Procedure

Observe which entities the interviewees can relate to and recognise. This can take expression of sudden recognition for processes when showing the map or that the new project is easy to fit into the map. Be observant when similar names are used to denote entities; that usually confirm information.

5.4.3 Analysis: Resolving Conflicting Information

Problem:	Collected data between projects can be of conflicting nature.
Solution:	Compare processes' output and input as well as objects and roles.
Output:	Follow-up questions for upcoming interviews or configurable characteristics

Procedure

Conflicting information will appear when trying to fit the newly introduced projects onto the process map created with the reference project. Processes, objects and roles will either be missing or have an incorrect configuration in relation to the recently introduced projects.

To detect all conflicts the naming and denoting of entities should also be considered. Entities can sometimes represent different things in different projects, even if they have similar or the same name. The opposite can also hold true, i.e. different naming and denoting represents the same entities, this is however not a conflict.

The identified conflicts can be mitigated in three ways:

a) Use the new information to draw the conflicting part of the map in a new way which is more representative for the organisation.

b) Formalise this conflict as a configurable parameter of the process map.

c) Create follow-up questions asked to the other projects within the study to bring more data to the issue.

5.4.4 Analysis: Conforming Information

Problem: The same processes, objects and roles may be referred to differently by different interviewees.
Solution: Examine interview discussions and find where the *same* subjects are discussed with *different* names.
Output: Updated process map with deleted, merged or renamed processes, objects or roles.

Procedure

During analysis of the diversity interview data it may be revealed that different processes and objects have a lot in common, but still are regarded as different entities. The naming can for example be specific for a field or context, but the subjects covered between them is exactly the same. These entities can be merged but their respective scope needs to be covered in the new merged entity. This usually means a slightly wider definition of the process in the end.

This reasoning also holds true for roles which sometimes share for example responsibilities and obligations. When merging roles make sure that they really are identical, otherwise responsibilities will be left unhandled. It may be that two persons' definitions of a role only corresponds to a certain degree.

5.4.5 Analysis: Responsibility Assignment Matrix (RAM)

Problem: The RAM is not configured for all processes or roles.
Solution: Create a baseline of responsibility types and assign roles to the RAM for all processes.
Output: Fully populated RAM

Procedure

This analysis method consists of two parts, first the "Responsibility Assignment Matrix" should be *configured* with responsibility-types, such as "Responsible" and "Informed", and then *populated* with the responsibilities in every process for each of the roles. A notation example is found in Figure 5.9.

The work of both configuring and populating the responsibility assignment matrix should be done when as many participants as possible have been interviewed to ensure the diversity of all possible roles and responsibilities are covered. It should then be validated in the final focus group.

Configuring the Responsibility Assignment Matrix

When the RAM is configured there are two kinds of responsibility-types, one covers the interaction role and process, and one covers role to role communication.

The role to process interaction concerns the value creation, often touching various kind of responsibilities. Ask yourself if recurrent types of responsibilities have been present during the study. If so, use them, otherwise bootstrap the matrix with the types: *Responsible (R)*, *Accountable (A)*.

The role to role communication types follow the same pattern. Ask your self if recurrent information flow have been present during the study. If so formalise this, otherwise bootstrap with the types: *Consulted (C)*, *Informed (I)*.

It is recommended to strive for having one RAM *configuration* for the whole process map. There may however be several RAM matrices on the map, having the same configuration, where each RAM contains the processes and roles for each core process. This is a trade off between effort for the readers of the map and usefulness as more RAMs increase cognitive load and therefore time consumption.

Populating the Configured RAM

After the configuration of the responsibility assignment matrix is complete the matrix shall be populated with the responsibilities of each role in every process. Start by going through every process on the map and list them on one axis in the matrix. Add all roles used in the map on the other axis of the matrix. Thereafter, based on observation during the study, determine the responsibilities each role has in every process, in a best-effort manner.

The RAM also aims to improve the overview of role assignment on processes. If a role has an assigned responsibility on the RAM, it shall also be attached on the related process on the map (see example in Figure 5.4).

5.4.6 Data Collection: Follow-up Interviews

Problem: Previous analysis shows that supplementary information is required to complete the map.
Solution: Ask specific questions to the correct sources.
Output: Text notes or updates on the process map.

Procedure

During activities in Step 2 of ECPM new issues are likely revealed were clarification is needed. These issues could be hard to resolve if not more data are collected, especially from the first reference project. This data is easiest collected by having short and specific individual interviews with the most appropriate interviewees. When issues are resolved the map is simply updated. To enable good preconditions for the follow-up interviews is it important to carefully take notes continuously during the analysis activities when issues arise.

5.5 ECPM Method Step 3 of 3: Validate the Refined Process Map

The goal of the third and last step is to validate the release candidate of the process map. The purpose is to review the result in order to ensure quality and to find out to which degree the process map can be utilised and applied at the recipient organisation. This step is not iterative like the two previous, but due to practical reasons it may be required to be split up and carried out in different parts with different focus.

5.5.1 Data Collection and Validation: Focus Group

Problem: The content of the process map needs to be cross-validate.
Solution: Perform a focus group which validates teh process map.
Output: Audio files, notes on the process map

Procedure

The focus group is a workshop in which ideally all previous interviewees and the organisation's stakeholders participate.

The focus group will both validate the map and elicit new information. The discussion should focus on the information which is already on the map, and not on eliciting new.

The goal is to moderate a discussion forcing cross-validation of the map between the participants. The discussion should at least cover the subjects: *Core Processes*, *Main Processes*, *Control Processes* and *Roles and RAM*. Further suggestions are completeness of the main processes, the flow of information and value between the processes, correctness of control processes, validation and verification of the roles' responsibilities.

As representatives from as many parts of the organisation should be represented is it an excellent opportunity to start discussions about the process owners role. As the process owners will take over the responsibility of maintaining the process map do they need to be well aware of the content.

5.5.2 Validation: Sanity Check of Included Processes

Problem: It is unknown if the process map covers all common activities related to software engineering.
Purpose: Match processes on the map to the process categories concluded in IEEE 12207-2008.
Output: A list of potential missing activities to follow up with more data collection and analysis.

Procedure

The IEEE 12207 standard [61] concludes 43 process categories which often are present within software related organisations. Categorise all processes drawn on the process map into these categories. Use the *purpose* for each category defined in the standard to help decide the sorting.

When all processes are sorted, analyse if some of the categories have none or very few processes present. Try to give a sound reason for that pattern based on previous knowledge of the organisation. The categories covers both main, support and control processes and as the process map is oriented towards value adding these will be dominating.

If no natural reasons could be found for the pattern is it suitable to follow up the issue with the studied organisation. Either is some aspect missing on the map or does the organisation not have the recognised capability.

5.6 ECPM Notation Language

The data collected during a process mapping study must be stored and presented in a standardised way to be useful. To achieve this is a lightweight notation language necessary to cover the use cases that the ECPM Method touches. The ECPM Notation Language, presented in the following pages, can of course be swapped out when using the ECPM Method given that the substitute supports the relevant information entities.

The notation language has a value refinement focus which not explicitly shows recursive patterns. It consists of a set of building blocks introduced in Section 5.6.1 which is combined according to a set of rules and principles explained in Section 5.6.2.

The ECPM Notation Language shares the terms and definitions with the ECPM Method, and is found in Section 5.1. The notation language is designed for generating process maps that are to be printed on size A0 paper. This makes it possible to view how all the processes are connected with each other without having to browse through a stack of papers or scroll in a large file on a computer.

As sources for the notation the models VPM, BPMN, IDEF0 and Rational Unified Process have had great influence to the design. [11, 62, 33, 63]. Also the article *The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations* [64] and industry experience from M. Lundgren [35] have been a firm support.

5.6.1 Graphical Building Blocks

The following two figures shows the notation of the basic process building blocks.

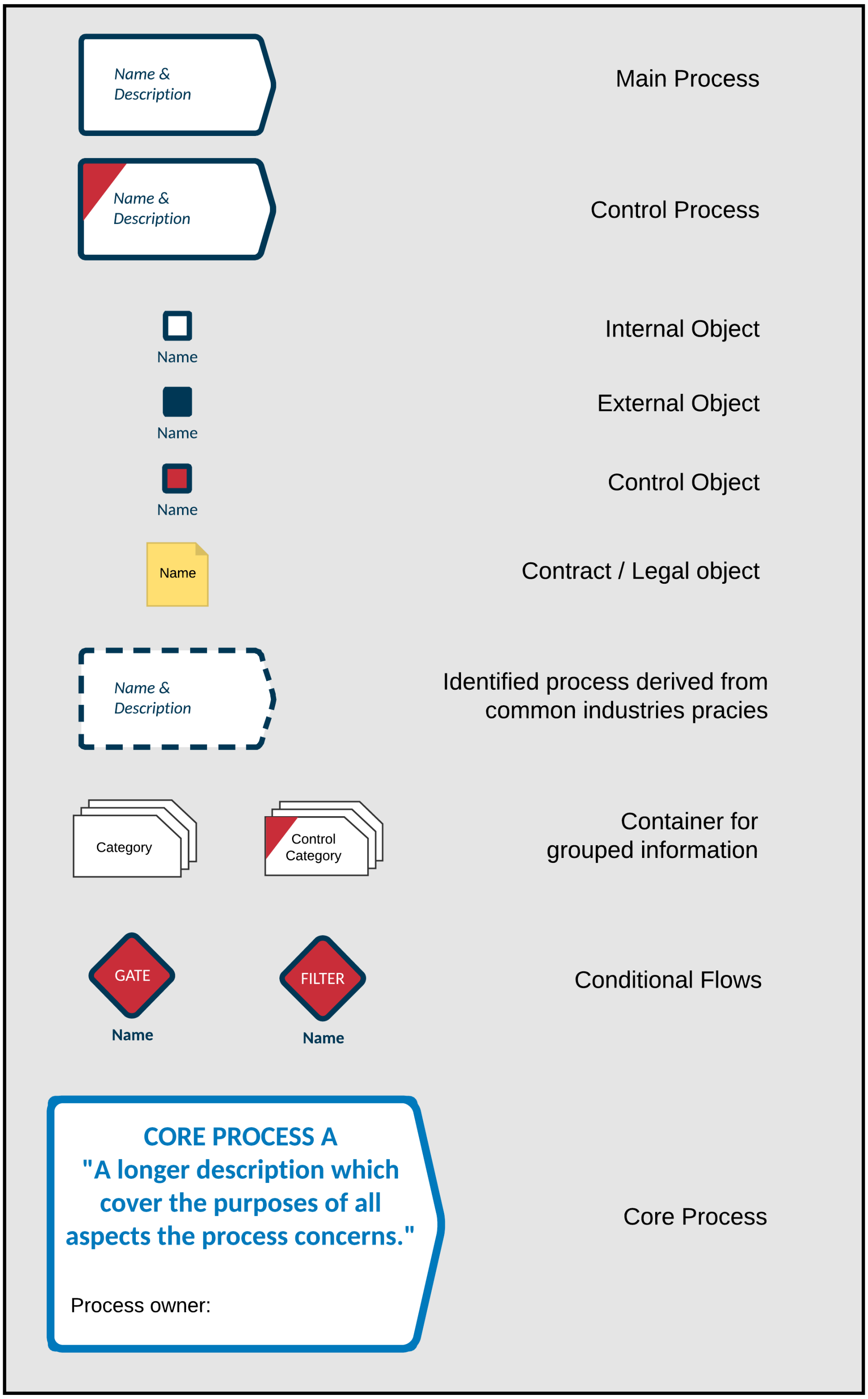


Figure 5.3: Building blocks with generic texts.

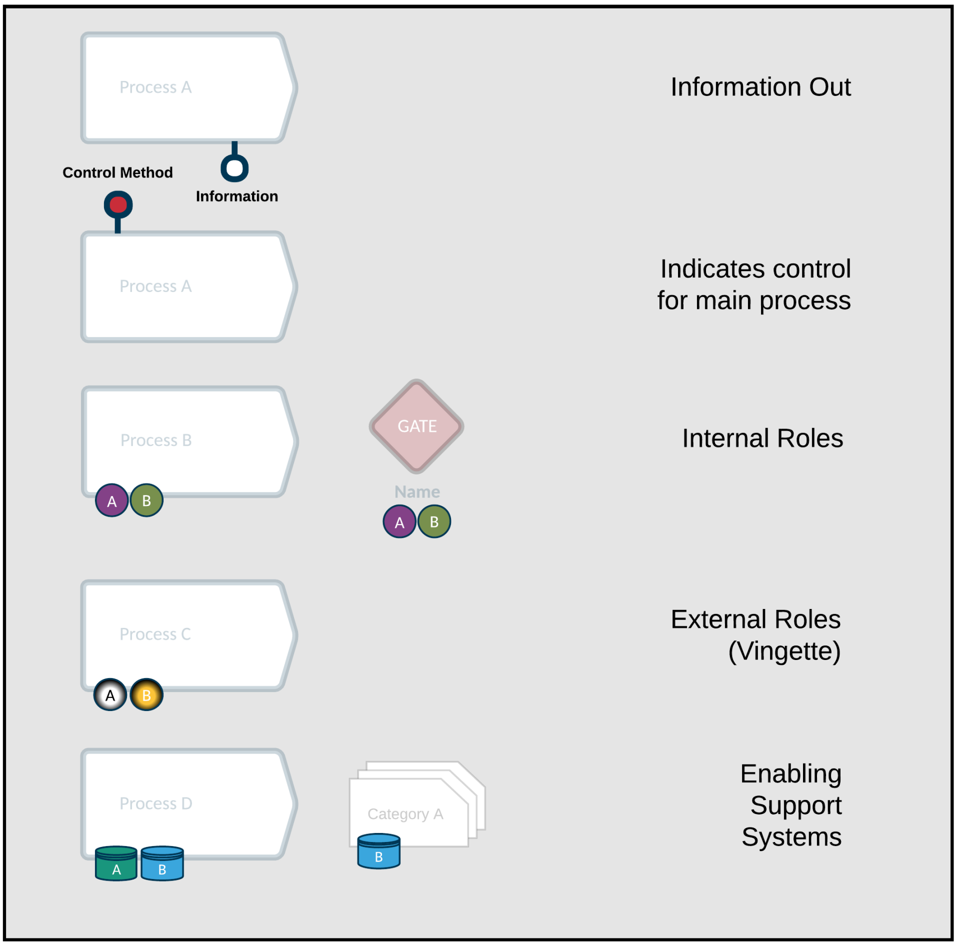


Figure 5.4: Attributes and properties with generic texts.

The symbols in Figure 5.4 are attributes that can be added on the process blocks (greyed out), and originally introduced in Figure 5.3. Multiple attributes can be combined and added to a single block, i.e. a process can both have "internal roles" as well as "information out" attached to itself.

The roles and support systems icons need to be complemented with a list explaining their meaning. Try to match the letter on the icon with the full-length name in the list; for instance could the role "Developer" form an icon together with "D" or "DE". Every icon should have a unique colour. External roles are staffed by the customer and are separated graphically with a vignette on the icon.

5.6.2 Drawing Principles of ECPM

In this section a set of figures demonstrates the key drawing principles in the ECPM Notation Language. Figure 5.5 illustrates different connection types between processes and the logic they represent, as well as the input and output objects. Figure 5.6 concerns the illustration of control mechanics. Figure 5.7 illustrates the different levels of the process map. Figure 5.8 illustrates the principle layout of a core process map. Finally, Figure 5.9 concerns the notation of responsibility assignment matrices.

Logic for Processes and Objects

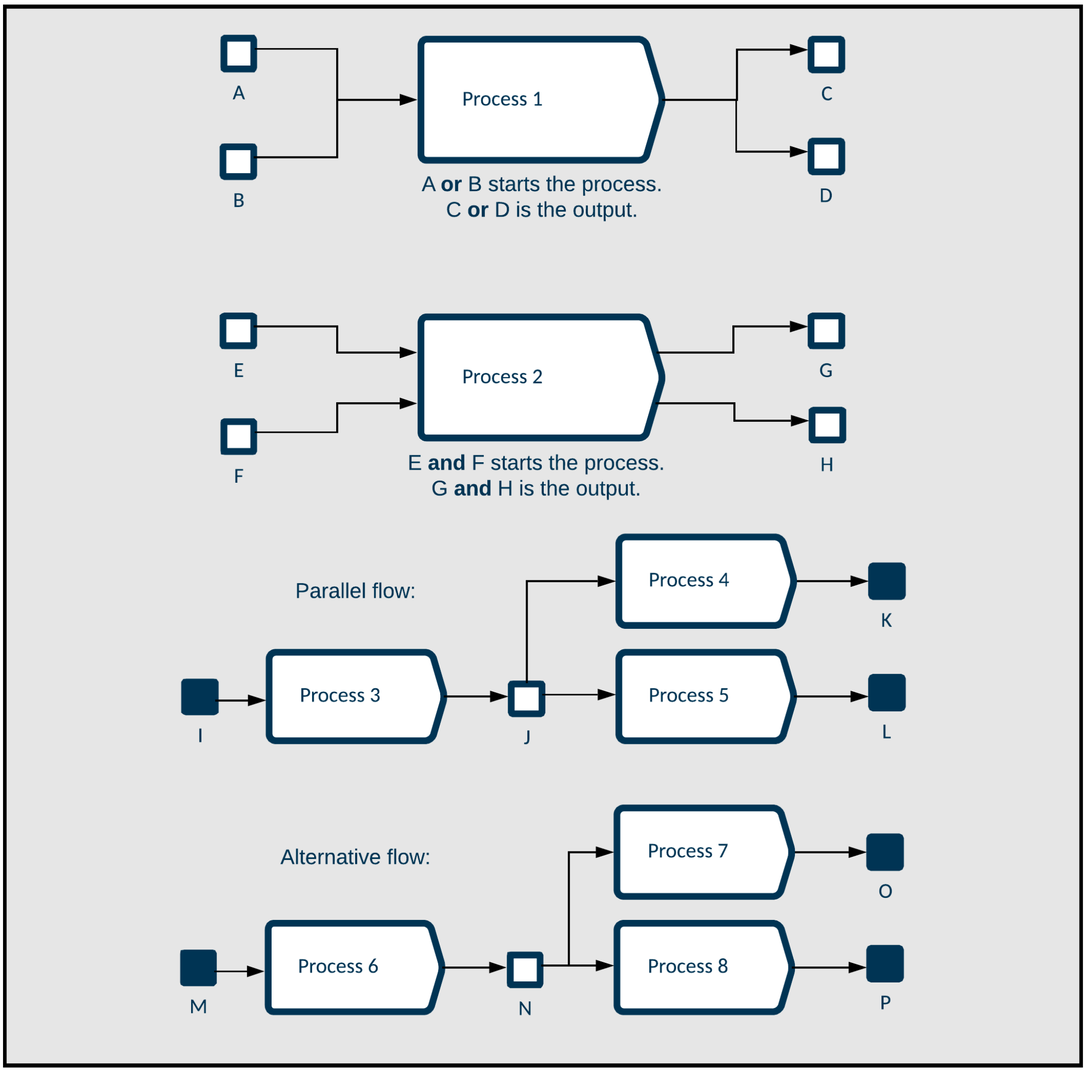


Figure 5.5: Four different connection types to processes.

A process will continuously run as long as it is being fed "input". A process is stopped by either ceasing its input feed or imposing a control method upon it. In Figure 5.5 inputs are marked by arrows and input need to be present on each *individual* arrow for the process to run. *Process 1* will start when A **or** B feeds it but *Process 2* needs both inputs E **and** F.

The objects which a process produces can either be **internal** and used within the value-chain (J) or **external** which means that they leave the value-chain to be received by a customer (K). Processes can produce a single object (J), multiple ones with choice (C,D) or strict multiple ones (G,H). Objects can have **parallel** flows (J) where the same object feeds multiple processes (*Process 4,5*) or **alternative** flow (N) where only one of several process is fed (*Process 7,8*).

Control Mechanics

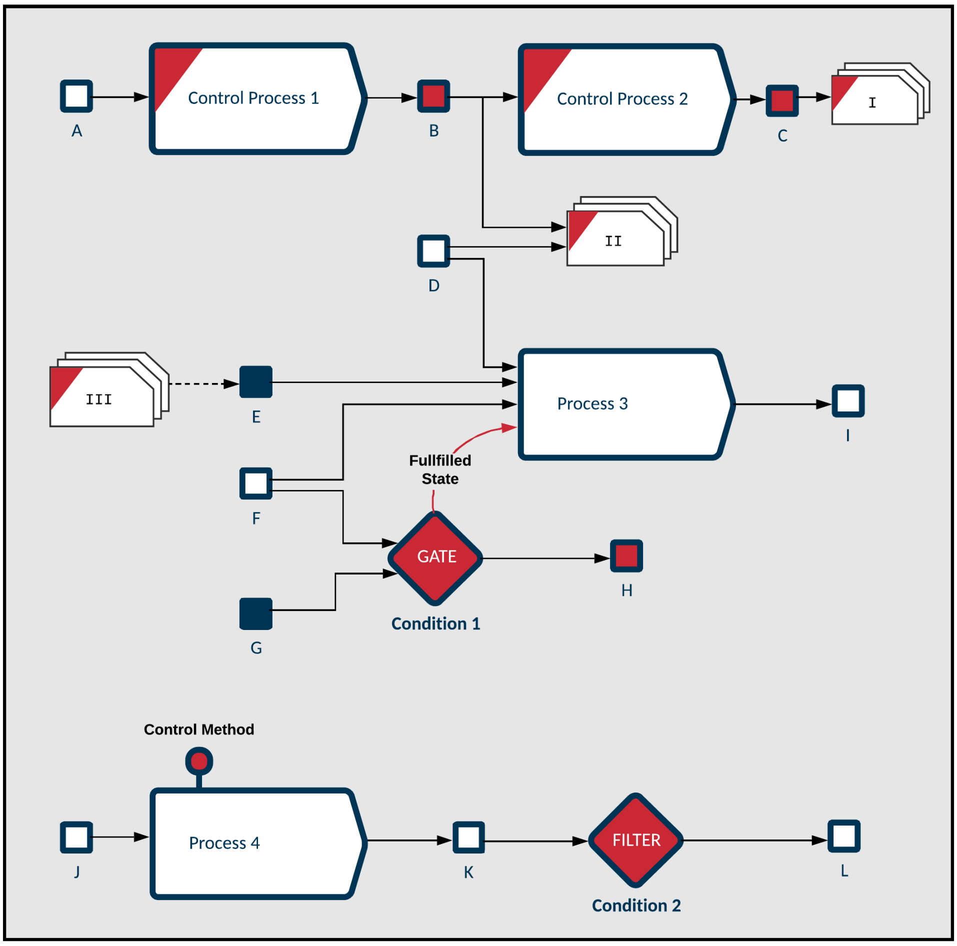


Figure 5.6: Overview of control mechanics.

Figure 5.6 above includes all control mechanics, illustrated in red, that is used in the notation language. Control features should be used with caution as they can be visually dominating and draw focus away from the main processes. The main principle is that control mechanics usually follows all other rules of the notation language. As can be seen with *Control Process 1 & 2*, they use the same logic as described in Figure 5.5 with one difference: a control process can only produce *control objects* (B). It can however be triggered by both *objects* (A) and *control objects* (B).

It is common that *control objects* are stored in control containers for later retrieval (C→ I), e.g. timeplan attributes are stored to a product backlog. Ordinary objects (D→ II) can also provide information to control containers which is used to execute control. Sometimes input to main processes are external objects (E) which are strongly related to or derived from a container (III), which can be illustrated with a dashed line. To mark that a process' execution is controlled or influenced by a control method a sign is attached as seen in *Process 4*. Finally there is execution conditions: Gate (*Condition 1*), which needs to be fulfilled before *Process 3* is allowed to be executed, and Filter (*Condition 2*), which means that objects (K) needs to achieve a certain level of characteristics or quality in order to pass the filter.

Multiple Levels

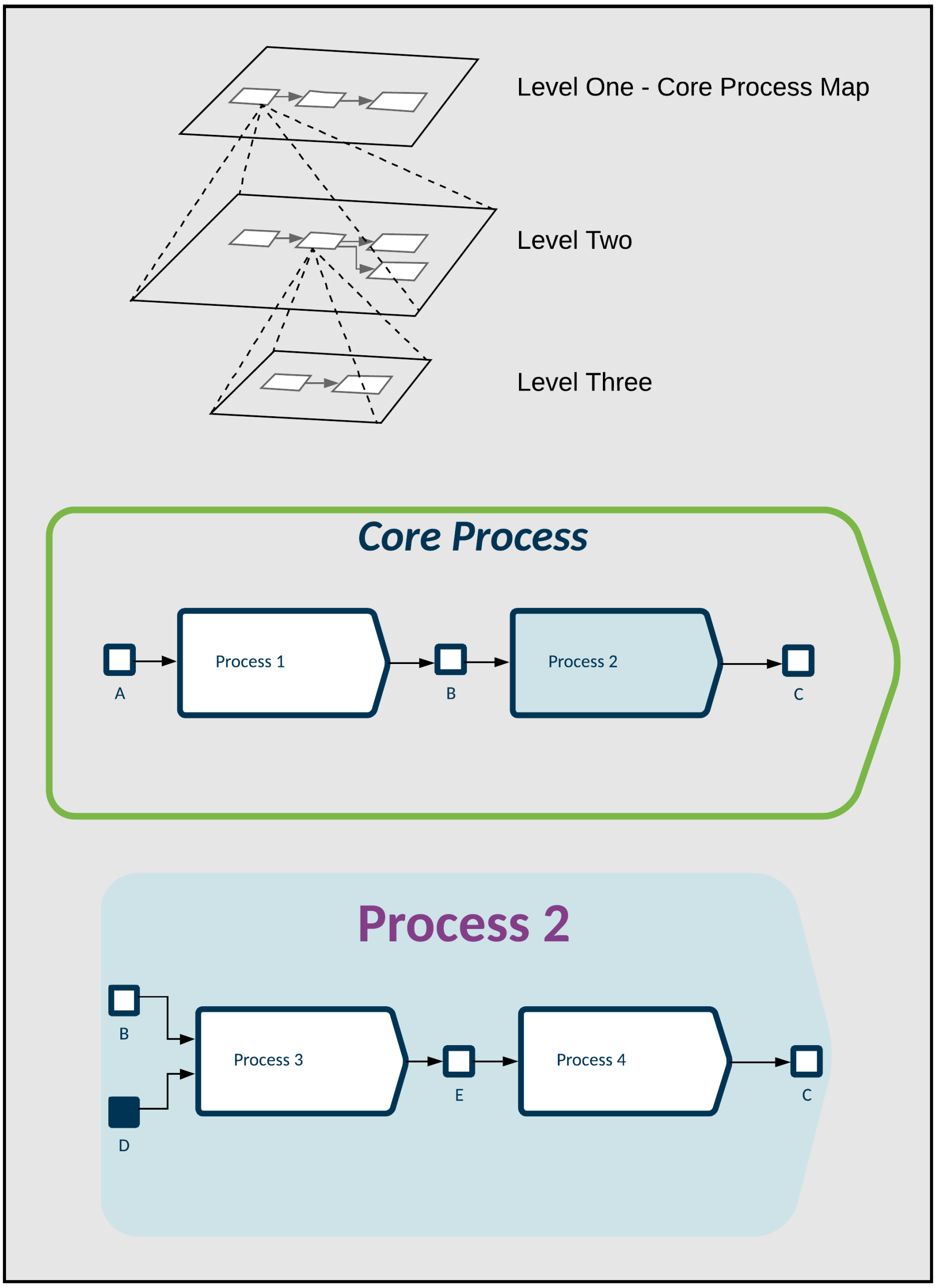


Figure 5.7: The multiple levels of a process map

The notation language uses a three-level system for drawing to improve the readability of the map. The top level consists of the *Core Process Map* which provides an overview of the entire organisation. The middle level contains all of the processes and the lowest level contains processes that are describing *how* things are done. Connections between levels are made through processes, and are illustrated with a shared colour code as seen on *Process 2* in Figure 5.7 above. The input that feeds the the upper level's process must be present in the one below (B) and output from the bottom process must be present above (C).

Core Process Map

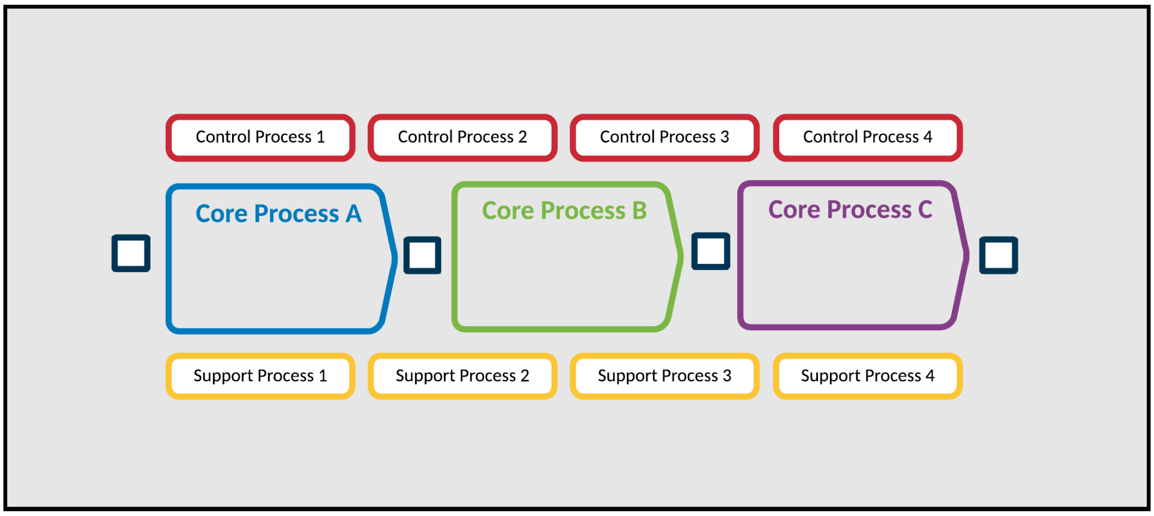


Figure 5.8: Principle layout of a Core Process Map

The top level of the process map is made up of the *Core Process Map*. This level aims to give an overview over all components that forms the process map. It has three distinct categories: *core processes*, *control processes* and *support processes and systems* as shown in Figure 5.8. The core processes shall cover the entire value chain from the first captured value to the last delivered value to the customer. When reading up on all core processes' purposes and their relation to each other it shall be clear what value the organisation provides.

Above the core processes all control processes are listed which influences any component within the process map. That is, if a control process is present in the process map it shall also be listed here with the purpose of giving a complete overview.

Lastly there are support processes and systems which are not printed out in the lower drawing levels and therefore it can be harder to sense what to include or not. A rule of thumb is that support processes should be visualised if they are essential for the processes, objects and containers. Examples of this are a specific issue tracking software or a recurrent service like maintaining a hardware test bed.

For practical drawing and printing reasons the list of roles and systems, as well as detailed explanations of control processes may be drawn in the vicinity of the *Core Process Map*, even though these objects are not related to the core process map itself. Refer to the process map in *Section 6.1: The Process Map* for an example.

Responsibility Assignment Matrix

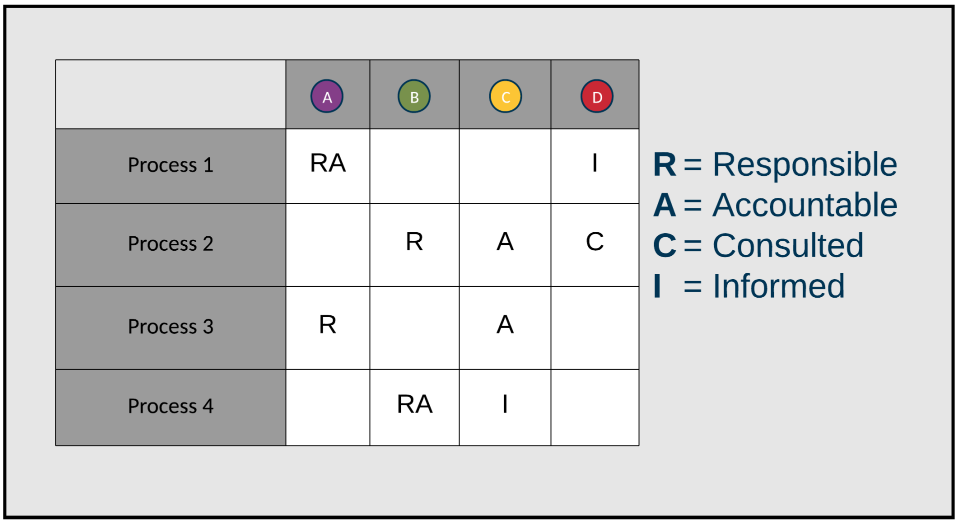


Figure 5.9: Example RAM with generic data

The Responsible Assignment Matrix is the notation's main tool for visualising interactions and communication between roles, and indirectly between people as they staff the roles. Figure 5.9 shows a generic matrix with the predefined responsibilities, in this example the *RACI* setup.

On the top of the matrix are all the roles present within the process map shown. To the left are all processes from the map which have at least one assigned role, e.g. *Process B* in Figure 5.4, present. In close proximity to the matrix should the available types of responsibilities be summarised as this is a configurable property between matrices. These responsibilities are then used to link roles to each other in the scope of a process. One cell can have multiple assignments or no one at all. For practical reasons experience suggests that it is suitable to have one matrix for all processes included in a single core process.

Chapter 6

Results: Applying ECPM at Combitech

This chapter presents the results from the application of the ECPM Method and Notation Language at Combitech. In Section 6.1 the process map created at Combitech is presented, followed by Section 6.2 which describes examples of the underlying key conclusions that formed the map.

6.1 The Process Map

The main result from the execution of ECMP at Combitech was the process map found at [65]. The process map was designed to be printed in a rather large size of 594 mm high by 4205 mm wide. As this thesis is printed in a considerably smaller physical size, a highly compressed overview version of the map is shown in Figure 6.1, with annotations A-F.

The process map includes a *Core Process Map* (A) which provides a high level organisational overview using a set of *Core Processes*. The Core Processes are illustrated both in the Core Process Map, and more detailed on their own (D1-D4). The Core Processes in turn contains the organisation's processes, with related *Responsibility Assignment Matrices* (E1-E4). Some of the processes are in turn drawn on an even more detailed level (F1-F2). The map also has a few *Control Processes* (B1-B4) which shows how work is managed, and a *legend* explaining roles and systems (C). All of the entities are defined previously in *Section 5.1: ECPM Terms & Definitions* and further explained in *Section 5.6: ECPM Notation Language*.

At Combitech five core processes were identified, out of which four were mapped in more detail: "Concept Exploration" (D1), "System Design" (D2), "Development" (D3), and "IT Operations" (D4). The core process "Development" had two of its processes expanded further: "Integration testing" (F1) and "Create and decide business solution for operations" (F2). These expansions were done as a lot of relevant data were collected by the ECPM Method in this area, but the map would have been too cluttered if all data were to be drawn directly in the core process.

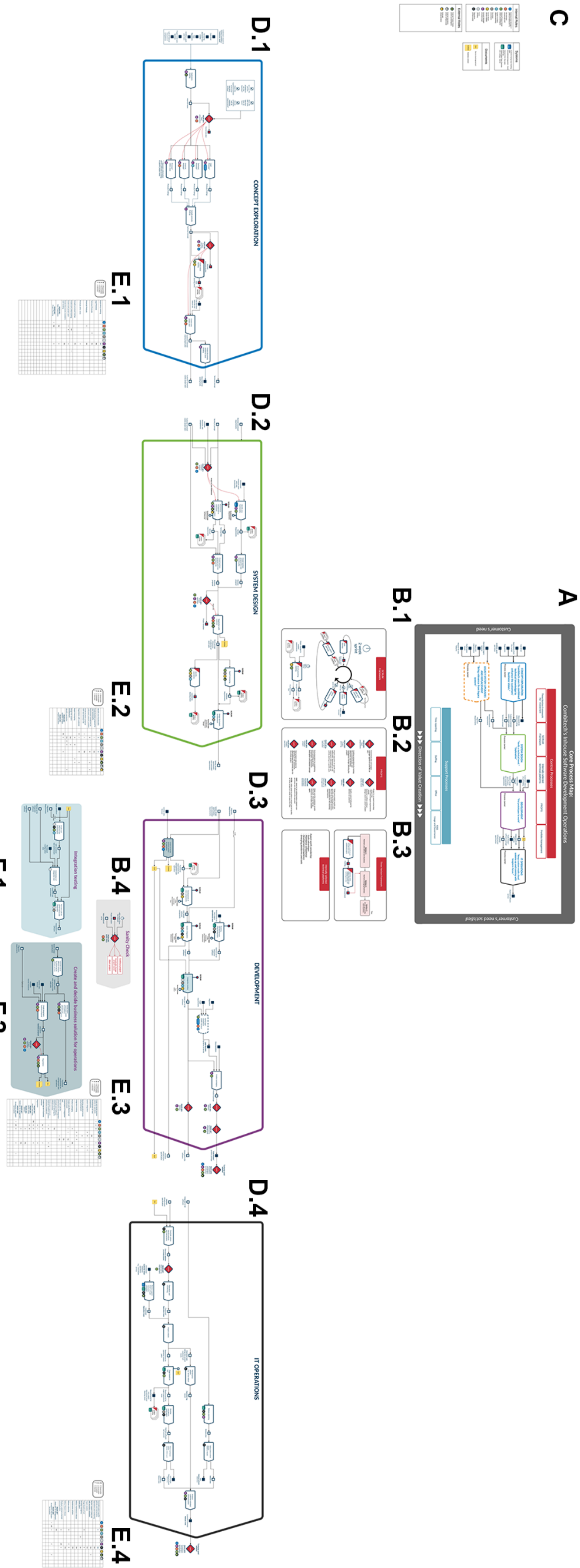


Figure 6.1: A scaled down version of the process map. A: *Core Process Map*, B: *Control Processes*, C: *Legend*, D: *Core Processes*, E: *Responsibility Assignment Matrices*, F: *Detailed Processes*.

The processes were mainly managed by three kinds of control processes: "Scrum" (B1), "PROPS" (B2), and finally "Three Horizon Framework" (B3). Of these three were only "Scrum" and "PROPS" directly interacting with the processes as explained in *Section 5.6.2: Control Mechanics*. One part of the "PROPS" control process (B2) includes a step of *Sanity Check*. This step performs a sanity check of a project after a certain amount of time, which in the studied case is related to the core process "Development", therefore this is drawn in a separate box (B4).

As a complement to the overview of the entire process map shown in Figure 6.1, two detailed examples are included in this report: Section 6.1.1 highlights how Scrum is modelled as a control process, and Section 6.1.2 displays how a operational issue is handled.

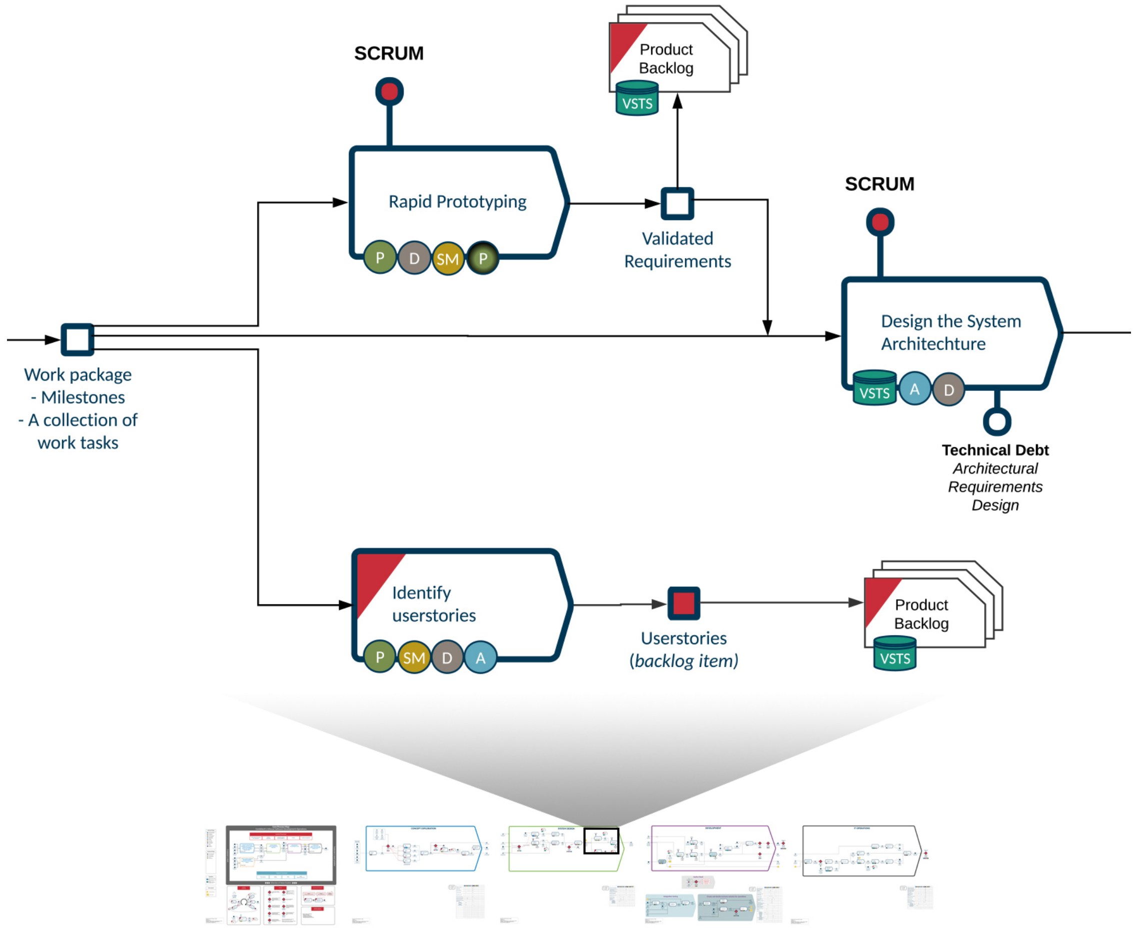


Figure 6.2: Early stage of the software development during the *core process* "System Design" (D2) where the *control process* Scrum influencing a lot indicated by the red markings.

6.1.1 Scrum Modelled as Control Process

In Figure 6.2 a zoomed-in part of the core process "System Design" (D2) demonstrates how the control process Scrum, which was commonly used at Combitech, can be illustrated on the process map. The two main processes "Rapid Prototyping" and "Design the System Architecture" are controlled by Scrum, i.e. the work carried out and the value produced are using the Scrum cycle and its practices.

The process "Identify userstories" on the other hand produces Scrum userstories categorised as control objects, in contrast to value, as they enables the Scrum cycle. These userstories are stored within a repository using Microsoft's *Visual Studio Team Services* (VSTS), a collaborative tool used for software development, as can be seen by the silo icon.

6.1.2 Detailed View of Operation Processes

The next example shows a triage procedure in the core process "IT Operations" (D4) which illustrates how deployment of software components to production are managed. This example shows among other things how an *object*, i.e. something produced earlier in the value-chain are used to control a process; in this case how a *Service Level Agreement* (SLA) controls the main process "Triage based on SLA" seen in Figure 6.3.

The process "Triage based on SLA" has two possible outcomes as seen in the top part of Figure 6.3: either critical requiring an urgent patch or non-urgent allowing the issue to be handled as a change request. The triage decision is two-folded; both technical and business aspects needs to be taken into consideration, where the Service Level Agreement (SLA) gives guidance in the later. Change requests originating from the triage process are handled earlier in the value-chain together with other change request such as feature requests. This is handled in the core process "Development", displayed at the lower cutout in Figure 6.3, and modelled as an *external object*, using a filled box, shown below as "Requirements for non-urgent operational issue" acting as input for the main process "Resolving operational issues using development". The object is used in the scrum cycle for control as well; notice how they are stored in the "Product Backlog" information container.

In all processes which deliver value for software put into production, i.e. almost all processes in the core process "IT Operation", played the role "operation" a huge part. This was notable compared to other core processes where the responsibilities were more evenly spread out between roles. A possible effect of this could be less communication and interaction within Combitech, as only a few persons with the same role performs most tasks.

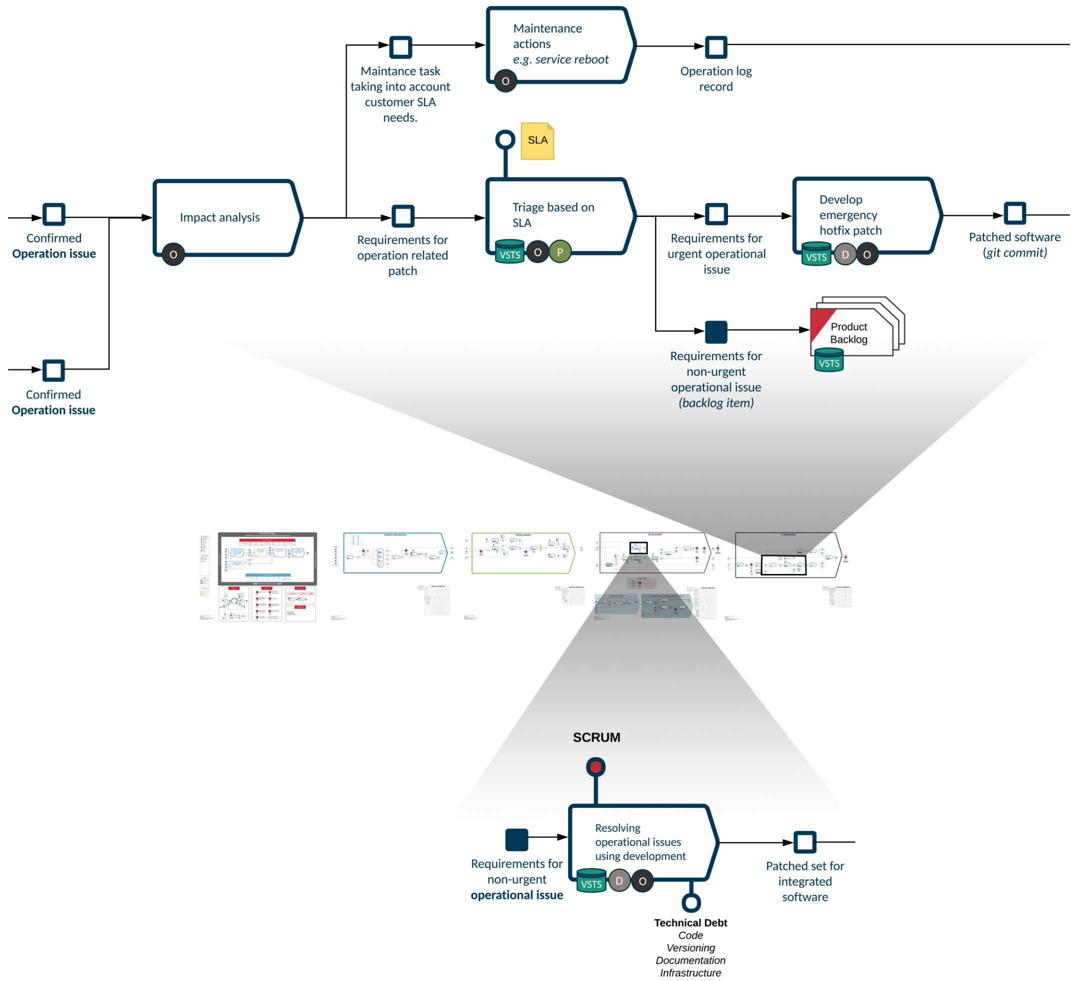


Figure 6.3: Top: Three processes found within the "IT Operations" core process partly displaying how an operational issue is handled. Bottom: A process which is part of the core process "Development" related to the operations pipeline described above.

6.2 Analysed Data Collected by ECPM

This section presents the key conclusions that was deduced from the interviews, informal meetings and the focus group, and used to establish the process map using the activities presented in *Section 5.2: ECPM Method*. These were used to form the process map shown in Figure 6.1. The key conclusions are divided into six groups depending on what topic they address.

The conclusions presented below are not needed to grasp the content of the process map; they provide traceability and improve the ability to review the ECPM method's validity and usability.

6.2.1 Elicitation of Processes

ECPM was built upon the idea of quickly making an inventory of all processes in its first step *Create*, to later correct and expand it in the step *Refine* with input from additional projects. Out of 38 processes were 34 identified in some form and can be traced back to the first step of ECPM. The definitions and names the of processes were updated later in the study to better reflect all the projects within the study.

Not all ECPM activities worked equally well for eliciting processes. One of the activities that performed very well, and supported identification of a majority of the processes, was the activity “Analysis: Follow the Process Flow to Find ‘Silent’ Processes”. This technique was used multiple times and found to be especially useful when analysing “Mapping of operation related processes” (*Interview 2*) and “How debugging works” (*Interview 3*).

Using a process map, even an early draft, wrong or incomplete, during the data collection activities helped the interviewees to point out what was unique with their working situation. The map also forced the interviewees to precise their statements and facilitated the work of structuring the data on the map.

Some activities in ECPM distinguished themselves to have a specific software focus and brought up new aspects not addressed by the other ECPM activities. The activity “Find Processes for Requirement Management” provided a structured way of sorting and exploring the first core process “Concept Exploration”. The software terminology and definitions were able to put words on many of the interviewees’ thoughts, and thus improved clarity. Information concerning “Organisation of pre-sale activities” (*Interview 1*) and “How sales responsibility works” (*Interview 2*) were revealed this way.

The approach of the software specific activity “Find Processes Generating Technical Debt” (*Section 5.3.8*), used to bring forward new processes by studying where technical debt was created, did not work as intended. It turned out that this was a difficult approach, but the activity did manage to find six cases of technical debt generation, which were marked on the process map.

6.2.2 Categorisation of Processes

During the study, processes were sorted into four categories: *Main Processes*, *Control Process*, *Support Process* and *Core Processes* in order to better distinguish what a specific work task contributed to. The detailed procedure of these categorisation activities are explained in *Sections 5.3.13 to 5.3.16*.

Categorising control processes was the most challenging of the four, as it often was hard to separate these from value adding processes. For example, whether Scrum falls within the value or the control category. Categorisation was ongoing during the entire study. An important breakthrough was the introduction of the “red notation”, used to visualise control processes differently from value processes. This enabled mapping of refinement in two parallel tracks, one concerning the value to the customer, and one showing how the organisation achieved better management capability through refined control information.

The distinction between value- and control information drawn on the process map helped the interviewees to relate value in the software context; i.e the interviewees could focus their story on either how they were controlled or what value they brought, adding missing data to the study.

The main topics of the five core processes were quickly identified in the first step of ECPM, but the exact limits between them was harder to decide. Processes adjacent to the borders were shifted between the core process multiple times as the refinement of the map continued.

The support process on the other hand were easy to specify, no cross validation and no conflicting information needed to be resolved. The support processes were finished early in the study and did not change later. The ambition of mapping was however lower for the support processes than it was for the other types, which probably made this work easier.

6.2.3 Object Identification and Categorisation

The strategy that was often used to find objects, i.e. the input and output from the processes, was to ask the interviewees what a specific process took as input and what the output form it was. This strategy produced a lot of information. The most of the used data came from the first three interviews.

Notable was that the opposite technique, to elicit processes from objects, were not that commonly used throughout the study. The interviewees tend to not bring forward information about objects during the earlier interviews. This lack of objects made it impossible to deduce processes from it.

Another issue was to clearly define and name objects. Several interviewees had their own perspectives on objects' definitions and hence needed these to be merged somehow. This was solely made by the authors due to time constrains. Not involving the interviewees in this work probably lead to lower correctness in the final result.

6.2.4 Role Identification and Assignment

After that the interviews in ECPM's first step was completed, was it possible to create a list of the roles that existed in the organisation. The initial list contained a few issues. One specific fault was the role "Product Owner", which handles requirements and roadmaps for a product range. Initially the authors thought that only one role existed to handle these topics, but apparently there was two types: one external product owner assigned by the customer that focused on the requirements of the customer's product, and one internally at Combitech that handled internal requirement towards the customer to improve efficiency.

These additional investigations caused some delays on the assignment of roles to main processes, which otherwise were assigned as processes were identified. This had in turn the consequence that the roles could not be validated until the very end of the study.

6.2.5 Configuration of Responsibility Assignment Matrix

In addition to visualising which roles that were associated to which processes, their specific responsibility compared to other roles were also shown on the process map. The reason for using the Responsibility Assignment Matrix (RAM) for this was to not pre-define what kind of responsibilities that were needed but rather adapt it towards the organisation's needs when their needs were uncovered.

At the first two interviews it became apparent that the information and communication aspect would need to be included within the RAM. Many interviewees pointed out information-flows, usually by email between persons or roles, within the scope of a *main process*. This was also cross-confirmed using the stakeholder mapping.

The first iteration of the RAM was only using one information related responsibility, *Informed*, which focuses on *pushing* information, e.g. "sending an email". This was however expanded later on to also cover a *pull* flow where information is needed from the organisation and its stakeholders. The final configuration, which also included *Responsible* and *Accountable*, was not set until the later part of the study with the purpose of getting an as complete data as possible to base the design-decision on.

6.2.6 Performance of Validation Activities

The validation activities performed during the study found contradictions in the process map as intended in the design. Examples of conclusions from the validation activity "Focus Group" are: "The configuration of the RAM matrix felt relevant and applicable for Combitech" and "Several of the attendees considered the first core process to be relevant, but they hadn't thought of its existence previously".

The validation activities could have achieved even better results if they would have been assigned more time, as the authors experienced these to be somewhat rushed. Strict moderation was needed to ensure all aspects were covered, instead of just a few aspects being discussed in too much detail.

The activity "Sanity Check of Included Processes" reviewed the map for missing types of software process. It worked as a good complement to other validation activities as it somehow counteracted the bias introduced in the study. The activity enforced a broader perspective to ensure not entire categories of processes had been overlooked and left out.

6.2.7 Study Time Usage

In Table 6.1 a list of what data collection activities that were carried out in each of the three steps of ECPM, and the time spent by Combitech in each of these activities, are shown.

Table 6.1: Overview of carried out data collection activities at Combitech and their respective time usage.

ECPM Step	Data Collection Activity	Combitech's Time Usage (h)
Create	Interview 1	1
Create	Interview 2	1
Create	Interview 3	1
Refine	Interview 4	4
Refine	Interview 5	2
Refine	Interview 6	2
Refine	Informal Meetings	10
Validate	Focus Group	8

Chapter 7

Results: Evaluation of ECPM

In the following sections the evaluation of the ECPM Method and Notation Language according to *Section 4.3: Evaluation of ECPM* is presented. Section 7.1 concerns the results from the evaluation interviews held at Combitech, while Section 7.2 presents the results from the authors' evaluation.

7.1 Evaluation Interviews

The design artifact described in *Chapter 5: Results: ECPM Method and Notation Language* was evaluated at Combitech using the process map, found above in Section 6.1.

The evaluation interviews contained questions which were of both qualitative and quantitative nature. All of the persons that took part in the ECPM activities also took part in the evaluation interviews, which were carried out some weeks after that the final process map was handed over.

A summary of the questions can be found in Table 7.1 together with the quantitative answers. The question are fully explained in Appendix B which also elaborate the exact meaning of the value for each question.

The quantitative answers are also visualised in a simple graph in Figure 7.1. However the number of respondents was in total 6, which has to be taken into account when reading the results.

Followed by the chart are the qualitative answers presented grouped together with in small topics.

Table 7.1: Overview of the evaluation interview question with quantitative results where applicable. '1' indicates the lowest and '4' the highest quantitative value. The quantitative results are visualised in Figure 7.1.

Question	**Mean**	**Low**	**High**
Q1: What purposes do you see for the ECPM map?	N/A	N/A	N/A
Q2: Do you consider a value-based perspective to be applicable and relevant for your role?	3.7	3	4
Q3: Do you experience the ECPM Process Map to be supportive for your work?	2.7	1	4
Q4: Do you think that the level of detail is well chosen?	3.5	3	4
Q5: Do you experience the ECPM Process Map to improve your understanding for your work in the big picture?	3.0	2	4
Q6: How well does the ECPM Process Map help you explain your work for your colleagues?	2.7	1	4
Q7: How complete in terms of recognised processes do you experience the ECPM Process Map to be?	3.4	3	4
Q8: Do you experience that the input and output objects between processes are generally correct?	2.5	2	3
Q9: Does the ECPM process map illustrate how your work is controlled and lead?	2.6	2	3
Q10: Do you experience the roles on the ECPM Process Map to be correct?	3.0	1	4
Q11: Do you think that it would be easy to reconfigure or update the ECPM Process Map in order to describe new projects?	3.2	2	4
Q12: How much time do you estimate one would need to grasp the content of the ECPM Process Map?	N/A	N/A	N/A
Q13: Has this research project taken any of your time, apart from the activities during which the researchers were present? If so, how much?	N/A	N/A	N/A

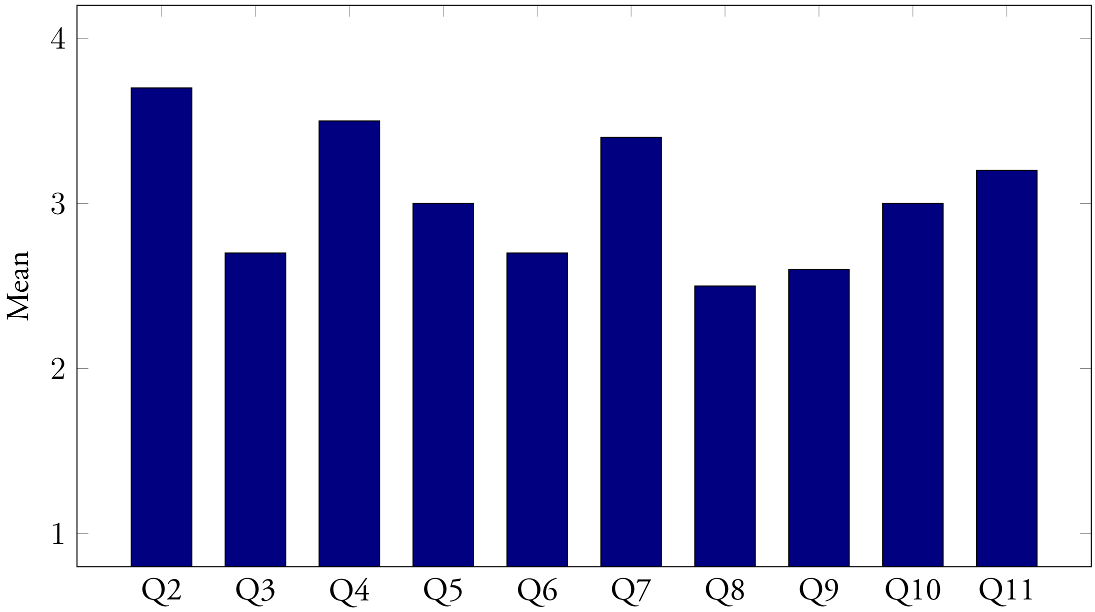


Figure 7.1: Quantitative results from evaluation interviews, where '1' is lowest and '4' highest score.

7.1.1 ECPM Method Performance

"Performance" was defined as how effectively the ECPM Method correctly identified processes and their associated input- and output objects, control structures, roles, etc. These were all visible in the map and therefore could the performance of the ECPM Method be evaluated indirectly using the process map.

The answers to Question 7: *How complete in terms of recognised processes do you experience the ECPM Process Map to be?* consider the method to have found most processes but to have missed an entire framework. The method also seems to have found more processes than the organisation was aware of. One participant answered *"I missed a particular framework, but apart from that most processes were included."*, while another said *"I found more processes in the map of the organisation than I was aware of"*. Another answer in line with this conclusion was *"I think there are some missing processes, but they are probably not critical. We don't know all our processes ourselves."*. Project-specific details were missing from the map, but that was rather a confirmation than a shortcoming. *"It covers most processes, but of course details specific to the projects needs to be added."*

The correctness of the input and output objects between processes was evaluated using Question 8: *Do you experience that the input and output objects between processes are generally correct?*. The answers tells that this is a weak area of the map: *"This is the part where the map is weak."* and *"If I would have studied the map more in-depth I would probably have found input and output objects that are incorrect or missing."*

The method's ability to find control process varies according to the evaluation. Question 9: *Does the ECPM process map illustrate how your work is controlled and lead?* has answers ranging from *"Very well."* to *"One framework is missing."*, where the later is in line with Question 7 above.

The interviewees mostly agreed on the roles in the organisation were properly identified.

Question 10: *Do you experience the roles on the ECPM Process Map to be correct?* was answered with statements such as *"Very good."*, *"You have covered it thoroughly."* and *"There are no description at all today so this is good."*. Some considered that a few roles were missing, but that the ones that were drawn on the map were correct, with statements such as: *"I think you covered it all. Maybe someone that was missing."* and *"Those roles that are on the map are correct but you have missed some."* However, there was one respondent that said that the map *"Doesn't describe the roles that well."*

7.1.2 Reactions on the ECPM Notation Language

The participants found the ECPM Notation Language to be easy to understand; one participant said, as an answer to Question 12: *How much time do you estimate one would need to grasp the content of the ECPM Process Map?*: "*Easy for an engineer to understand how the model work, approximately 30 minutes. Another 30 minutes to understand what the map illustrates.*" Other participants referred to the guide (selections from *Chapter 5*) that was presented together with the process map, and said it was easy to understand the notation language after reading it. Some respondents would like to have a structured walk-through with the authors describing every part for half a day.

Given the answers to Question 4: *Do you think that the level of detail is well chosen?* does it appear that the participants overall were happy with the level of details that the notation illustrates, for example did one participant say that: "*Well balanced, given that the map isn't the only source of information.*", and "*Good that the map has three different levels of detail, so that different persons can select a suitable level of detail for their role.*" Some requested more information regarding roles, and some asked for little less details "*Slightly too detailed. Too many special cases. Would have been more generic and easier to explain if it was less detailed.*" A conclusion was given by one of the respondents, who said: "*Well balanced. It is up to each project to set the level of detail.*"

7.1.3 Reception of the Process Map

In addition to evaluating the performance of the ECPM and the process map, was the interviewee´s opinions of use-cases for the process map collected. Their opinions on the relevancy of a value-perspective were also collected.

Use-cases for the Map

Some of the most central answers from the evaluation are what use-cases the group saw for the map. Question 1: *What purpose do you see for the ECPM map? e.g. what problems do you think it may solve?* addresses just that, and was answered with the following:

- "A base for the work of change in the company."
- "Find problems."
- "Find ways to change for the better."
- "A base to find what processes that are parts of standards and established methods, and what is our own ideas."

- "To show how we work both for us internally and for clients."
- "A tool when introducing new employees to provide an overview of how we work, and to show what parts the new employee should work with."
- "Illustrates and provides a base for discussion of how handovers between processes should be made."

The participants considered that the map was supportive of their work. Question 3: *Do you experience the ECPM Process Map to be supportive for your work?* was answered with: "Yes, it lets me focus my energy on solving my tasks.", but that it is required to keep the map constantly updated: "Yes, but it is changed constantly and needs to be updated". Some respondents found the map less useful and thought it to be useful "On certain occasions". The authors interpret that as some respondents have roles in the organisation where a process map is useful, and some don't.

Further, the respondents mostly agreed on that their understanding of their work in the big picture was improved by the map, confirmed by the answers to Question 5: *Do you experience the ECPM Process Map to improve your understanding for your work in the big picture?*. For example: "Yes. I have many roles and these are reflected in the map." and "I get a very good understanding. Where and when does the other get decision-support. What forum exists."

Question 6: *How well does the ECPM Process Map help you explain your work for your colleagues?* had very varying answers. Some regarded it to be very useful, and some didn't find it useful at all. One person working with the phases before the developers take over most of the workload found it useful to explain his work: "It makes the developers understand that there is work done before their work begins. It makes them understand that someone made an informed decision that a project should start, so they don't have to question it."

Relevancy of Value

The answers to Question 2: *Do you consider a value-based perspective to be applicable and relevant for your role?* indicates that most of the participants considered the value perspective from which the map is built upon to be correctly chosen as a common denominator. "That is probably a good aspect", "It is very suitable" and "Yes, but it is hard to explain, especially to developers. It is, however, the value we deliver that we get paid for."

Maintaining the Map

The nature of the process map requires that it accurately reflects reality by being updated regularly. It is therefore important that it is easy to incorporate changes in it. The question *Question 11: Do you think that it would be easy to reconfigure or update the ECPM Process Map in order to describe new projects?* got varying answers, ranging from "Certainly" to "Seems hard". Many interviewees found it hard to give a precise answers as they have only done minor hypothetical changes to the map, i.e. "Seems so, but don't know as I haven't done it before."

Time Consumption

By summarising the length of the data collection activities, as seen in Table 6.1, was one metric on time consumption found. The authors was interested in two other metrics as well; the time the participants considered that was needed to grasp the content of the process map, and if the participants had spent any time on the process mapping project other than what the authors were aware of. The later was of interest as one of the goals of ECPM was to take very little of the participants' time.

The participants answers to Question 12: *How much time do you estimate one would need to grasp the content of the ECPM Process Map?* were varying between less than half an hour, to a week. Examples of answers was:

- "Easy for an engineer to understand how the model work, approximately 30 minutes. Another 30 minutes to understand what the map illustrates."
- "One week, as some parts needs to be repeated and explained."
- "After reading the guide I could understand the map. Then 10-15 minutes to verify that I understood the map correctly."
- "It would be hard in an hour, but a structured walk-through would facilitate understanding. Maybe half a day."

Question 13: *Has this research project taken any of your time apart from the activities during which the researchers were present, if so how much?* had answers that indicated that no more time was spent for the purpose of the study, but that the participants had studied the map out of curiosity and discussed it with colleagues for a few hours in total.

7.2 Evaluation by the Authors

The authors' experiences from applying ECPM at Combitech are presented below. The statements are their objective observations of the properties and performance of ECPM and the process map.

7.2.1 Method

The authors' evaluation of the method part of ECPM focuses on the practical questions of time consumption, scalability and missing features in the method.

The method consists of 26 different activities. Not all of the activities have to be applied to create a map, and most of the activities don't have to be carried out in a particular order.

It was important to early in the project draw a process map of the studied organisation. If a map wouldn't have been drawn early in the mapping session, there would probably have been lots of unrelated small sketches and notes, and practically no overview of the organisation to relate succeeding information to.

Time Consumption

Creating process maps for organisations is time consuming and thus expensive. To make it easier to plan and allocate resources, estimates of the method's time consumption is presented below.

- Based on the authors' estimates and the summation of the length of the audio files from the interviews, the studied organisation have spent approximately 30 hours on this project. This number is supported by the evaluation interviews.
- The authors estimate that 300 hours have been spent on the execution of ECPM at Combitech, including data collection, analysis, validation, and drawing of the process map. This estimate aims to give a cost of re-running ECPM to produce an additional process map, why efforts covering construction of ECPM and report writing are excluded.
- The first iteration of the process map could be drawn after an estimated 80 hours of data collection and analysis.
- Step one, *Create*, consumed most time for the studied organisation, consequently the staff involved in these steps need to have high availability.
- Setting time aside for familiarisation of the projects' context was important for the authors in order to improve understanding of the interviewees.

Scalability

This ECPM execution was carried out by the two authors, who both participated in all activities. If ECPM should be applied to a larger organisation or at several offices of one organisation it might be necessary to increase the numbers of researchers.

- Step 2 and 3 of ECPM, which mainly consists of follow-up interviews and validation, are probably possible to be assigned to several researchers. The work in these two steps investigates details based on an already existing process map, which makes it easier to communicate and share findings between the researchers.
- Both authors were present in all activities, e.g. data collection, analysis, creation of the map etc. It is hard to tell to which extent it is possible to scale the method by splitting up the work between more researchers and still maintain quality, or assigning a single person to carry out interviews. For instance, during the interviews one researcher was more prominent and led the interview, while the other intervened when the discussion didn't follow the interview guide, asked follow-up questions for clarification, and took notes.
- ECPM is designed to gather *all* participant from *all* included project at a final focus group. The maximum limit of participant within the focus group will therefore indirectly limit the amount of possible projects ECPM can cover in a study.

Shortcomings

During the execution a few missing features or capabilities of ECPM were identified. They are not to be seen as limitations of the study.

- ECPM does not have a structured method on how to assess and manage bias based on a specific interviewee's personal viewpoint and role.

- ECPM lacks a specific method to verify the result of input and output objects entering or leaving processes, which also is proven to be a shortage according to the evaluation interviews.

- ECPM does in itself not provide methods or capability of traceability from collected data, and analysis, to objects drawn onto the process map.

- ECPM does not have a method that consolidates and conforms multiple definitions of a role.

- ECPM does not include any formal requirements on previous knowledge or professional background to be able to use the tool. With different backgrounds of the researchers may the process map's focus shift.

7.2.2 Notation Language

The authors' evaluation of the ECPM Notation Language focuses on its capabilities and shortcomings.

The notation language consists of 20 unique components all describing different information. The notation uses a set of 10 rules for describing logic and flows. The ECPM notation language focuses on the relationship between main and control processes. If control processes are omitted from the process mapping another notation language might be preferred.

The case study was conducted on an organisation which had no previous process related model to start from. That meant that no direct comparison of quality of the process map and time consumption for the studied organisation can be performed.

Capabilities

A selection of noteworthy capabilities of the ECPM Notation Language are presented below:

- The representation of control processes enables for example the Scrum framework, which was central in the studied organisation, to be well integrated into the map.

- The notation language gives the option to integrate a wide range of IT support system into the map.

- The logic used for connecting processes to illustrate the value flow is unambiguous and stringent.

- The researchers assessed that it took approximately 30 minutes for the participants in the study to grasp the notation language.

- The information in the notation can be filtered by using different levels of drawing and hence reduces the cognitive load on the reader.

- The researchers experienced that all data collected during the study was possible to draw using the notation.

Shortcomings

During the execution of ECPM the authors noted three deficiencies of the notation language:

- The notation lacks features to show localisation, e.g. if an organisation has two sites there is no possibility to tell which processes that are performed where.
- The notation have some difficulties to clearly point out cyclic patterns.
- The notation language is primarily designed to handle the concept of value. In a context of strict physical flows like logistics or databases another notation could be better.
- The value perspective was not self-explanatory but the concept model introduced in *Chapter 5: Results: The ECPM Method and Notation Language* eased the learning process substantially.
- The notation expanded to a map with huge physical dimensions.
- The information density of the process map was by average increased almost 9 times between the core process map and level two; respectively 3 times between level two and three.

Chapter 8

Discussion

In this chapter the results presented in the three previous chapters will be discussed in Sections 8.1 to 8.3, followed by a discussion regarding threats to the validity of the findings in Section 8.4.

8.1 Creating a Process Model Efficiently (RQ1)

The authors suggests that the work of creating a process map for an organisation that develops software could be organised according to the ECPM Method, described in Chapter 5.

The *Core Process Map*, a part of the process map, provided an overview over the entire organisation on a high level and was created early in the process mapping project at Combitech. The Core Process Map was partly used as a tool to ensure that no parts of the organisation would be overlooked. If an area of the organisation would have been errantly excluded early in the study, it could have lead to major rework later to correct the process map. This strategy worked very well as the required amount of rework of the map was very low, and no part needed to be completely redrawn. The researchers' time could therefore be spent on the "correct" activities, i.e. improving the quality of the process map. The quality of the final process map was considered to be high, so with the foregoing reasoning implies this that the work was carried out efficiently.

As the development of ECPM was driven by Combitech's problems is ECPM inevitably biased towards their needs. By reviewing the ECPM activities is it clear that only a few of them specifically handles *objects*. By comparing this to the number of mentions of *processes* in the activity descriptions, an evident shift of focus towards processes appears for ECPM. This is in line with the results from the evaluation reviews, which points out that the precision of the objects on the process map is lower than the precision of other entities on the map.

As a consequence of focusing the data collection towards creating an overview which covers a wide scope, details were not always collected. One example are *objects*, i.e. the input or output of processes, which can have a lot of detailed characteristics that are useful when creating quality measurements.

At Combitech the authors identified that there were a lot of information regarding software tests available and ready to be used to created good measures. However, the ECPM Method needs to be expanded to handle this data collection.

In close relation to quality measurements is the handover between processes, also depending on detailed information. A good handover is characterised by the responsible for each process agreeing on a set of qualities an object needs to have before being handed over and sent forward in the process chain. The ECPM Method does not have a specific activity to deal with handovers in its current state, and needs to be expanded to support that feature.

It is interesting to consider the ratio between the amount of time spent by the studied organisation and the time spent by the researchers. A company undertaking a process mapping study may have very limited possibilities to set aside time for their software developers to let them participate in interviews and workshops (i.e. internal time). It is not easy to quickly add more programmers in a project either, as onboarding takes time. A solution could be to use a process mapping method which instead shifts the time consumption towards the researchers or management consultants (i.e. external time).

All in all, the authors spent 300 hours creating the process model for Combitech using the ECPM Method and Notation Language. This does not include the time spent on developing the ECPM Method and Notation Language in itself, just the process map which Combitech had asked for. Combitech spent 30 hours on creating the process model, which gives a ratio of 1:10 between the time usage of the studied organisation and the researchers. It should be noted that this comparison is only based on one application of ECPM, and is thus weak in its nature.

8.2 Illustration Capabilities of the Notation Language (RQ2)

The Process Map (Section 6.1), drawn using the ECPM Notation Language, have used value refinement as a common denominator to illustrate how Combitech works.

An important question to ask is if ECPM produces a too generic map which only shows a theoretical mapping of value refinement, which is hard to utilise in a real setting. One way to look at this is to examine how dominating the value refinement is compared to other information entities on the process map and how they integrate into each other.

By studying the process map the authors consider that three areas enabled the map to be practically useful and actually support the execution of day to day tasks, namely illustrations of *control processes*, *responsibilities* and *communication*.

The ECPM Notation Language's way of illustrating *control processes* and *main process* in parallel with the same symbol set, but with the control processes separated by red markings, made it innovative. This enabled the map to show refinement of both value (denoted "objects"), e.g. software a customer needs, along with refinement of control information (denoted "control objects"), e.g. a user-story used to control what software to develop. The authors have so far not seen this concept in any other existing notation language.

This capability of the ECPM Notation Language was notably useful when modelling the agile method Scrum used by Combitech. By the definition of ECPM does Scrum in itself not provide any value as the output of its activities concerns leading and managing of software processes, which is a *control process*. There was still though a need for illustrating refinement of control information, e.g. Scrum's pre-defined artifacts and processes for altering them such as sprint planning and backlog grooming. In the notation language control objects and processes were marked with a red label to enabled this illustration.

Responsibilities, which is the concept of using humans to ensure activities and goals are reached, are commonly present in the process map. The *processes*, acting as common denominators, does all have roles assigned to them to illustrate which roles that have responsibilities in a particular process. This means that almost everything shown on the process map is closely related to a human and thereof reducing the risk of having orphan information floating around. This is even expanded further to show specific kinds of responsibilities for different roles (developers, marketing etc.) using the Responsibility Assignment Matrix (RAM).

The illustrations of communications in the ECPM Notation Language does take up less space compared to other entities and focuses solely on the interaction between roles which are also modelled using the RAM. This only enables the ECPM notation to display intra-process communication, that is communication within a process, as the RAM is built and sorted around processes. To illustrate communication using the RAM is it necessary to configure it for this by including the suitable responsibilities for it, such as "Input" and "Informed". In the processes map that was produced in the study at Combitech was the RAM configured as "RACI", i.e. "Responsible", "Accountable", "Consulted", and "Informed", with the *informed* and *consulted* parameters displaying communication.

The notation's weaker capabilities of displaying inter-process communication, i.e. communication between processes, is not as effective as for intra-process communication, which is clearly a drawback. The notation can display outgoing information from processes but as seen in the process map is this rarely used.

One desirable feature for a process map mentioned earlier is to formalise how the handover sequence between processes shall be illustrated. The handover is crucial when responsibilities are carried over and communication is a huge part of this. The ECPM Notation Language does not support visualisation of handovers nor have the needed methods for collection of appropriate data to do so. It is clear that other methods should be better suited to elicit those perspectives, for example the FLOW framework [34].

Another perspective taken into account during development of ECPM was how cognitively demanding the resulting process model would be to understand. This is an important factor as a process map needs to be widely adopted in an organisation to be useful, something that would be difficult if the map was difficult to interpret. The evaluation interviews showed that the interviewees' estimation of how much time it would take to grasp the content of the process map had a large variation: from half an hour to up to a week.

Reasons for this could be whether the map is used for *understanding* or to *support realisation*; two goals which requires different depths of details and understanding. Another reason is the fact that all interviewees had different roles, and thus needed different type of information.

ECPM Notation Language Compared to Other Models

During the study the authors came in contact with a wide range of process models and frameworks. The similarities and differences between these models and the ECPM Notation Language are of interest to study to see what makes the result unique. The comparison was done by summarising and assessing fourteen other models used or interacted with throughout this study, presented below in Figure 8.1.

The comparison was focused around a set of five capabilities and topics which were central for the description of the projects at Combitech. The assessment of each models was done towards what the authors perceived as the most prominent capability and focus of each model. The five capabilities were:

i) Showing who is responsible and accountable.

ii) Illustrating where value-refinement occurs.

iii) Visualising human communication.

iv) Illustrating time dimension of projects.

v) Showing and communicating how things are controlled.

When reviewing the comparison chart the ECPM Notation Language stands out and is unique in the aspect of merging the *communication perspective* with *value refinements*. It is also worth noting that ECPM compared to the other notations covers a lot of aspects, i.e. it could somehow be classified as a generalising method.

This was also reflected in the evaluation interviews done at Combitech, where the respondents used words such as "base" and "common platform" to describe their idea of the map's purpose. Another conclusion from the evaluation interviews is that the interviewees considered the map to be reasonably complete.

These results also go hand in hand with ECPM's aimed applicability and focus of revealing relationships between main and control processes which creates an overview.

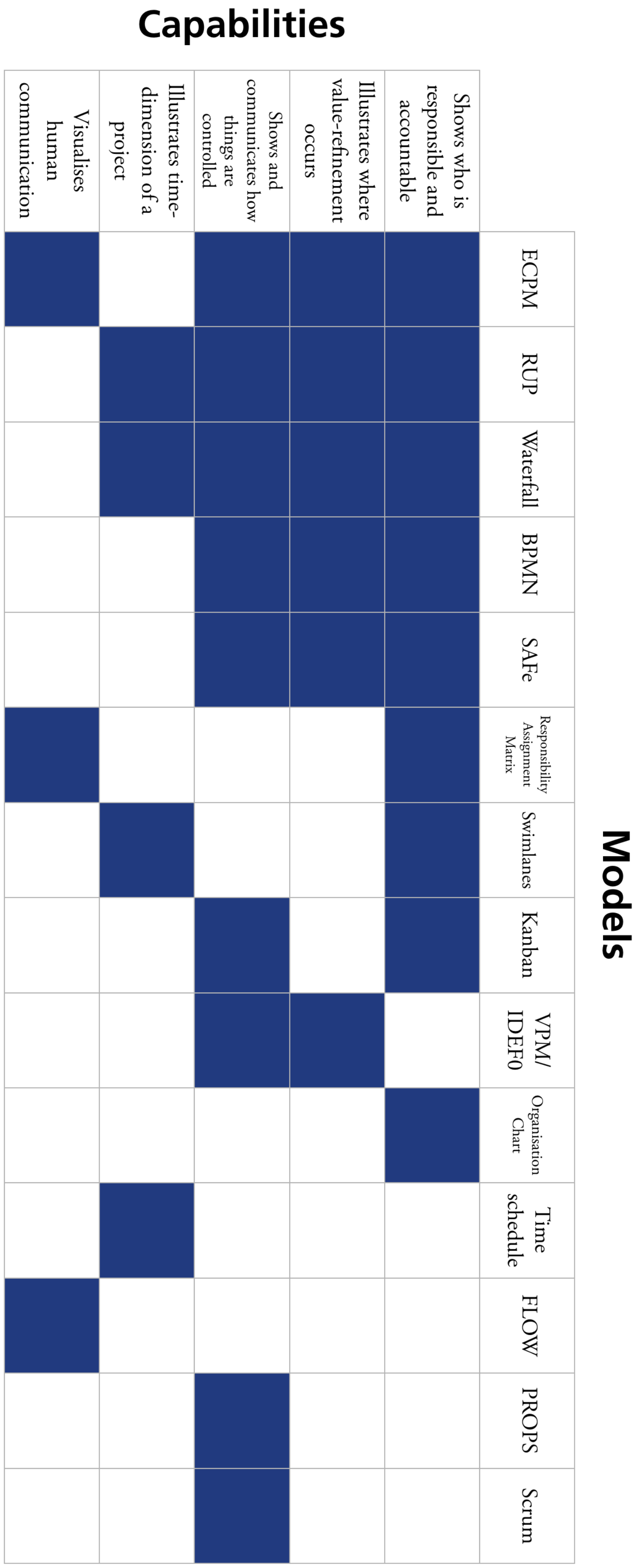


Figure 8.1: Comparison of different models' capabilities.

8.3 Configurable Properties of the Process Map (RQ3)

For a process map to be broadly useful, especially in the Combitech case where the process map aimed to transfer knowledge and standardise a way of working, does it have be adaptable to different contexts. To enable flexibility the notation introduced nine different configurable properties, presented in Table 8.1. These properties were found by analysing the finished ECPM Notation Language and Process Map together with Combitech at the end of the study.

It is hard to exactly tell if the notation is flexible enough for Combitech's needs without actually following up the long-term usage. The answers from the evaluation interviews' indicates however that Combitech upon receiving the process map had the capability to reconfigure the process map for their needs. Combitech did however express that some additional guidance would be necessary for this to work practically. The authors found that the detailed description of the ECPM Notation Language in Chapter 5 would be providing such guidance.

The conceptual model "Perspectives of a Project" (see Figure 5.1) was developed to explain the relationship between value refinement, time usage and how work were controlled. This models was also experienced to be supportive by the participants of the study when they were asked to do a reconfiguration of the map, as it provided context to the terminology and the relation between control processes and main process.

It is relevant to ask what the consequences of a reconfiguration would be if Combitech decided to change the process map while it was in use throughout their organisation. A reason for reconfiguration could for example be a new business case that has emerged for Combitech. In order to indicate a level and scope of which a process map property could be reconfigured without major implication on the rest of the organisation, the authors have concluded four scopes seen in column "Scope/Level" in Table 8.1.

All of these process map properties can be reconfigured but will affect the organisation differently. For instance, the more employees involved in a reconfiguration of the map, the more resources and effort will the reconfiguration take to complete, which in turn will lead to more interrupts in the production.

Table 8.1: Identified configurable properties that can be customised depending on context.

Scope/Level	Process Map Property	Description / Comment
Enterprise Reconfiguration	Core Processes and Main Processes	It is possible to add new blocks of value refinement when for example the organisation is expanded.
	Support Processes	These processes are often centralised in the organisation due to cost savings and therefore is it on this level of reconfiguration the changes must be done.
Process Map Optimisation	Roles	Roles can be assigned to individual processes throughout the process map.
	RAM	The Responsibility Assignment Matrix can have separate configurations between different core processes or other distinct parts of the process map.
	Technical Debt	Reducing technical debt is often related to how work is done, e.g. process structure, which is common for the entire process map.
Project Customisation	Control Processes	What kind of control mechanics can be swapped without altering the value refinement described by the process map.
	System Support	Support systems are often strictly connected to involved stakeholders' systems and therefore are they required to be configurable.
Runtime Adjustment	Project Start & End	The scope of what value are delivered to customers can vary during a project therefore where to entry and exit the process map can be flexible.

The scopes presented in Table 8.1 are further explained below:

Enterprise Reconfiguration: Redesign of the organisation or creation of either an entirely new process map or new parts to an existing map.

Process Map Optimisation: The level of configurations that shall be applied onto all projects. The role "process owner" is suitable for handling these kind of reconfigurations.

Project Customisation: This regards properties specifically changed from the process map during project initiation to describe project-unique needs.

Runtime Adjustment: Changes that are made during a project execution that doesn't invoke changes to the process map but still influences the outcome.

Another possible configurable property that wasn't covered by the process map but was discussed at Combitech was localisation. When reading the process map there is no possibility to tell if and where multiple groups are working with the same process. In the context of the studied case the result should be sufficient as the study had a single site focus, but it is reasonable to think that using the notation for a higher company level such as the entire business area would show this drawback.

8.4 Threats to the Validity of the Thesis

According to ECPM's scope of applicability in Section 5 was ECPM designed to be applied to an organisation that develops software development as a service. The focus of consulting is not that prominent in neither the method or the notation language. This means that probably could ECPM be used by any software developing organisation, consultancy or not.

As the method contains many activities that are of software type may the method not be useful for companies that doesn't work with software. For the method to be useful in other industries would it probably need activities to be added or removed. The notation language is more general and should be applicable in any industry with minor modifications.

8.4.1 Construct Validity

This thesis uses the produced process map, and experience collected while producing it, to review the ECPM Method and Notation Language. The most important design decision influencing the construct validity is the fact that all evaluation of data have been done through the *process map*, e.g. validation activities and post study interviews. The data collected with the ECPM Method is not validated with other means then by its own product, i.e. the process map.

The evaluation is in turn solely based on the initial reception; and therefore are just how well the process map provide "understanding" for the organisation, and not how well the map "supports the execution"-aspect measured. To precisely measure that effect, more time and resources would be needed to evaluate usage over an extended time period to see the effects of the implementation of the process map produced by ECPM, something that was out of scope of this thesis. However, the process map was thoroughly validated using a wide range of roles within the organisation, contributing to the cross-validation of the result.

The experiences from the usage of the method, important for the evaluation, are only drawn from the creators of the method. This actuality was of course hard to change due the nature of the master's thesis, but ultimately should the ECPM be used by a third party and observed, and interviewed, while doing so.

8.4.2 Internal Validity

The interviews didn't always follow the interview guides strict. There were primarily two reasons for this: either was the answer too interesting or too vague and needed to be followed-up with more questions, or was the interview guide just used as "inspiration" (primarily in interview types 2 and 3) and not a strict series of questions. This makes it harder to re-create the results. The follow-up questions depends on the interviewer's competence and experience. This would make it difficult to scale the method with several interviewers in a reliable way, as the competence needs to be the same for all interviewers. In this study the same interviewers performed all interviews.

In the end of each interview the interviewee was asked for a suitable next person to interview, as the authors considered the members of the team to best know who that might be.

8.4.3 External Validity

The answering of the research questions was limited to a single case study at Combitech. It was further limited to not only just one of Combitech's offices, but to a specific team. Even further was it limited to three of the team's projects, and not all members and roles of the team participated.

With that in mind is it obvious that the created process map can not be generalised to any given software organisation, or even to any given team within Combitech. It is reasonable to ask if the three studied projects represent the range of projects Combitech intend to use this model for. Fortunately, that is not the intention of the study, as it was to develop ECPM. A new map needs to be drawn for any new organisation that should be described.

The ECPM Method and Notation Language, however, should be possible to apply for any software developing organisation.

Combitech was used as representative of a software development company. Considering that Combitech are active in several different industries and fields, the company should be efficient in doing so. Combitech is still just a single example, and ECPM would probably benefit by being improved through use at other organisations.

Chapter 9
Conclusions

9.1 Purpose

For a consulting company to be successful within the software industry is it necessary to be able to quickly adapt to its clients' varying workflows and organisational structures. Combitech needed an overview of their organisation in order to be able to do so. A process-oriented overview, which uses value as a common denominator, allows for both software- and other types of business processes to be illustrated in the same model, thereby enabling the illustration of several types of value-creation both internally and in the clients' organisation.

9.2 Method and Results

The Design Science Research framework was used to iteratively create and evaluate the "ECPM Method and Notation Language", which constituted the main result.

By applying the ECPM Method and Notation Language at Combitech a process map describing their value-creation was generated. The methods of the ECPM Method included formal interviews, informal meetings and focus groups as data sources.

The artifact ECPM was then evaluated both directly and indirectly using the process map, i.e. the result of the developed method. The evaluation involved both participants of the study through formal interviews, and the authors.

The map was drawn digitally but was designed to be printed on paper with the size of 594 mm high by 4205 mm wide.

9.3 Research Questions

RQ1: How could the work of creating a process model for an organisation that develops software be organised efficiently?

The authors suggest that the work could be organised according to the "ECPM Method". The method consists of three steps: "Create", "Refine" and "Validate", all with associated activities for data collection, analysis and validation. During the first step a process model was created based on a reference project. In the second step was the process model refined by including projects similar to the reference projects in order to improve the model's accuracy. Finally, in the third step was the process model validated to be representative for the studied projects. The process model was then ready to be applied onto the organisation where it was applicable.

RQ2: What can a value-based process model for an organisation that develops software look like and specifically which entities of information should be included?

The ECPM Notation Language created together with the ECPM Method was used to form a process map which exemplified what a value-based process model could look like.

The process map contained information about refinement of value, i.e. the methods to satisfy customers' needs, the engagement and responsibilities of the work force while delivering value, and by which means the production of a certain value is lead and controlled.

RQ3: Which properties of a process model could be reconfigurable to ensure higher versatility?

The configurable map properties found in the study were: what core, main, support and control processes to include, what kinds of roles to be illustrated in the map, what types of responsibilities available in the responsibility assignment matrix, which support systems that are used, and where the process map starts and ends.

9.4 Contributions

The research was considered to be successful in helping Combitech with their experienced issues, which was indicated by the evaluation. A selection of the specific contributions to Combitech are presented below. There are also several contributions to academia.

9.4.1 To Combitech

The main contribution to Combitech was the process map, but the ECPM Method and Notation Language was also provided as a handbook so that Combitech can create new process maps and update the existing one created in this study. The ECPM method assists Combitech in conducting an inventory of the processes in their organisation based on projects within a studied group. The inventory can be done to help lead the organisation towards repeatability of a previously executed successful reference project.

The process map shows an overview of the entire value creation from the first contact with a potential client, to the delivery of operating software. The process map can therefore show how Combitech is working and how it is organised, both for clients and internally.

An overview of this type can be used for several things, for example illustrating how the organisation is controlled, what support systems that exists for a specific task, where technical debt is generated, and dependencies between processes. The authors also suggest that the process map could be helpful in preparing the organisation for certification according to the ISO 9000 standard of quality management systems.

9.4.2 To Academia

The most prominent contributions to academia are the two evaluated design science artifacts "ECPM Method" and "ECPM Notation Language".

The introduction of the "value" concept as a common denominator made it possible to value software components such as technical debt and requirements, and place them into a managerial supply chain perspective.

In *Section 5.1: ECPM Terms & Definitions* the authors connected the value-creation perspective with the time dimension of a project.

A comparison between several process mapping notations is found in Section 8.2. This comparison could be used to help researchers choose the appropriate notation for other mapping problems.

The interview guide that was written for the initial design of ECPM is based on and extended from the interview guide in the FLOW framework by Schneider [34]. The interview guide could be used independently of the ECPM Method when information regarding a software company's processes should be elicited.

9.5 Further Research

The authors suggests that the main focus of the further research should be to continue to improve the ECPM Method and Notation Language by finding another case company to execute the research method on.

This work should ideally be done over an extended period of time so that the model's qualities of supporting the organisation can be evaluated as well, something that couldn't be done properly in this study due to time constraints.

To enable the process map to better support every-day work the map needs to contain measures. To be able to display measures on the process map the ECPM Method needs to be expanded with data collection activities for handovers between processes and for objects' detailed characteristics.

As it took a lot of time to draw and update the process map manually would it facilitate if a drawing tool that supports automation is either found or developed. The tool should let the user edit the processes as objects with properties in a database, and delegate the drawing and rendering of the process map to a computer.

The problem with automating the drawing activity is that the drawing activity in itself can be considered to be an analysis method. The reason is that the person drawing the map is exposed to the information more thoroughly by working with it.

Another issue of automating the map generation could be that the selection of information considered to be particularly important, and therefore should be highlighted in the process ma,p is based on the subjective decisions by the person drawing the map.

An important capability that a tool for automating map generation should provide is tracking of the incremental changes so that a specific piece of information found in a data collection activity can be traced to an object on the map.

In addition to the printed process map would it be useful to find or develop another digital tool which let the reader *interact* with the map. For instance could the tool have features for showing all the processes and responsibilities assigned to a selected role, and allow the user to change configurable properties of the map to let it better describe a particular project.

Bibliography

[1] Pierre Bourque and Richard E. Fairley, editors. *SWEBOK: Guide to the Software Engineering Body of Knowledge*. IEEE Computer Society, Los Alamitos, CA, version 3.0 edition, 2014.

[2] Per-Hugo Skärvad and Jan Olsson. *Företagsekonomi 100*. Liber, 2011.

[3] Horace Bookwalter Drury. *Scientific Management: A History and Criticism*. Columbia University studies in the social sciences. Columbia University, 1918.

[4] Hans Torin. Combitech presents year-end report for 2017, 2018.

[5] Klara Palmberg. Exploring process management: are there any widespread models and definitions? *The TQM Journal*, page 203, 2009.

[6] Jan Edgren and Per-Hugo Skärvad. *Nätverksorganisationer : outsourcing, partnerskap och nya organisationsformer*. Liber, Stockholm, 2. edition, 2014.

[7] Thomas H. Davenport. *Process innovation: reengineering work through information technology*. Harvard Business Press, 1993.

[8] Geary A. Rummler and Alan P. Brache. *Improving performance: how to manage the white space on the organization chart*. Jossey Bass Business and Management Series. Jossey-Bass, 1995.

[9] August-Wilhelm Scheer Mark von Rosing, Henrik von Scheel. *The Complete Business Process Handbook: Body of Knowledge from Process Modeling to BPM, Volume I*. Morgan Kaufmann, 2014.

[10] Ken Schwaber and Jeff Sutherland. *The Scrum Guide*. November 2017.

[11] Anders Ljungberg and Everth Larsson. *Processbaserad verksamhetsutveckling*. Studentlitteratur AB, 2012.

[12] John P. Kotter. *Leading Change*. Harvard Business School Press, 1996.

[13] Peter G.W. Keen. *The Process Edge: Creating Value Where It Counts*. Harvard Business Press, 1997.

[14] Anders Ljungberg. Process measurement. *International Journal of Physical Distribution & Logistics Management*, 32(4):254, 2002.

[15] James Beiman Norman Fenton. *Software Metrics, A Rigorous and Practical Approach*. Chapman & Hall CRC Innovations in Software Engineering and Software Development. CRC Press, 2015.

[16] Walker W. Royce. Managing the development of large software systems: concepts and techniques. *Proc. IEEE Westcon, Los Angeles*, pages 1–9, August 1970. Reprinted in *Proceedings* of the Ninth International Conference on Software Engineering, March 1987, pp. 328–338.

[17] Kent Beck, Mike Beedle, Arie van Bennekum, Alistair Cockburn, Ward Cunningham, Martin Fowler, James Grenning, Jim Highsmith, Andrew Hunt, Ron Jeffries, Jon Kern, Brian Marick, Robert C. Martin, Steve Mellor, Ken Schwaber, Jeff Sutherland, and Dave Thomas. *Manifesto for Agile Software Development*. 2001.

[18] Kent Beck. *Extreme Programming Explained*. Addison-Wesley Educational Publishers Inc, November 2004.

[19] David J. Anderson and Andy Carmichael. *Essential Kanban Condensed*. Lean Kanban University Press, 2016.

[20] CMMI Institute. *Capability Maturity Model Integration*. `https://cmmiinstitute.com/`, 2019. Last accessed 31 January 2019.

[21] Randy A. Steinberg, David Cannon, Lloyd Vernon, Lou Hunnebeck, and Stuart Rance. *ITIL Service Lifecycle*. AXELOS, 2011.

[22] *ISO 9000:2015 Quality management systems – Fundamentals and vocabulary*. Standard, International Organization for Standardization, Geneva, CH, September 2015.

[23] *ISO 9001:2015 Quality management systems – Requirements*. Standard, International Organization for Standardization, Geneva, CH, September 2015.

[24] INC Scale Agile. *Scaled Agile Framework*. `www.scaledagileframework.com`, 2019. Last accessed 31 January 2019.

[25] Scott W. Ambler and Mark Lines. *Disciplined Agile Delivery (DAD): A Practitioner's Guide to Agile Software Delivery in the Enterprise*. IBM Press, 2012.

[26] Soren Lauesen. *Software requirements: styles and techniques*. Pearson Education, 2002.

[27] Ilene Burnstein. *Practical Software Testing: A Process-Oriented Approach*. Springer Professional Computing. Springer New York, 2003.

[28] Wayne A. Babich. *Software Configuration Management: Coordination for Team Productivity*. Addison-Wesley, 1986.

[29] Tom Milligan. *White paper: Better Software Configuration Management Means Better Business: The Seven Keys to Improving Business Value*. IBM Rational Software, 2003.

[30] Daniel Moody. The physics of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, Nov 2009.

[31] Patrice Briol. *BPMN, the Business Process Modeling Notation Pocket Handbook*. Lulu Press Inc., 2008.

[32] Tobin J. Lehman and Akhilesh Sharma. Software development as a service: Agile experiences. *2011 Annual SRII Global Conference, SRII Global Conference (SRII), 2011 Annual*, page 749, 2011.

[33] Information technology – Modeling Languages–Part 1: Syntax and Semantics for IDEF0. *ISO/IEC/IEEE 31320-1:2012 (Adoption of IEEE Std 1320.1-1998)*, pages 1–120, Oct 2012.

[34] Kai Stapel, Kurt Schneider, and Eric Knauss. Beyond documents: Visualizing informal communication. *2008 Requirements Engineering Visualization, Requirements Engineering Visualization, 2008. REV '08*, page 31, 2008.

[35] Trivector LogiQ AB Michael Lundgren, Senior Consultant & Business Area Manager. Private Interview, April. 04 2018.

[36] PMI. *A Guide to the Project Management Body of Knowledge (PMBOK Guide), 4th Edition.* Project Management Institute, 2008.

[37] Taho Yang and Chiu-Wen Chen. An incentive pay system for project management based on responsibility assignment matrix and fuzzy linguistic variables. *Expert Systems With Applications*, 36:12585 – 12591, 2009.

[38] Parvez Mahmood. Khan and Kaleem A. Quraishi. Impact of RACI on Delivery and Outcome of Software Development Projects. In *2014 Fourth International Conference on Advanced Computing Communication Technologies*, pages 177–184, Feb 2014.

[39] Mattias Mellhorn and Elizabeth Bjarnason. Improving goal communication with information flow maps and distances. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 276–283, Sept 2017.

[40] Kai Stapel and Kurt Schneider. Managing knowledge on communication and information flow in global software projects. *Expert Systems*, 31(3):234 – 252, 2014.

[41] Soren Lauesen and Otto Vinter. Preventing requirement defects: An experiment in process improvement. *Requirements Engineering*, 6(1):37–50, Feb 2001.

[42] *ISO/IEC25010:2011, Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models*. Standard, International Organization for Standardization, Geneva, CH, March 2011.

[43] IEEE Recommended Practice for Software Requirements Specifications. *IEEE Std 830-1998*, pages 1–40, Oct 1998.

[44] Mark A. Nash and Shiela R. Poling. *Mapping the Total Value Stream: A Comprehensive Guide for Production and Transactional Processes*. 2009.

[45] Thomas H. Davenport. *Process Innovation: Reengineering Work Through Information Technology*. Harvard Business School Press, 1993.

[46] Chris Beams. *How to Write a Git Commit Message*. `https://chris.beams.io/posts/git-commit/`, 2014.

[47] Herbert H. Clark. *Arenas of Language Use*. University of Chicago Press, 1993.

[48] Dewayne E. Perry, Nancy A. Staudenmayer, and Lawrence G. Votta. People, organizations, and process improvement. *IEEE Software*, 11(4):36–45, July 1994.

[49] Allen H. Dutoit and Bernd Bruegge. Communication metrics for software development. *IEEE Transactions on Software Engineering*, 24(8):615–628, Aug 1998.

[50] Raquel Andrade Barros Ouriques, Krzysztof Wnuk, Tony Gorschek, and Richard Berntsson Svensson. Knowledge management strategies and processes in agile software development: A systematic literature review. *International journal of software engineering and knowledge engineering*, 2018.

[51] A. César C. Franca Fabio Q. B. da Silva, Catarina Costa and Rafael Prikladinicki. Challenges and solutions in distributed software development project management: A systematic literature review. In *2010 5th IEEE International Conference on Global Software Engineering*, pages 87–96, Aug 2010.

[52] Alan Hevner, Salvatore March, Jinsoo Park, and Sudha Ram. Design science in information systems research. *Management Information Systems Quarterly*, 28:75–105, 03 2004.

[53] Karl Cox Hiyam Al-Kilidar and Barbara A. Kitchenham. The use and usefulness of the ISO/IEC 9126 quality standard. *International Symposium on Empirical Software Engineering, 2005.*, Nov 2005.

[54] An investigation of the weaknesses of the ISO 9126 international standard. *2009 Second International Conference on Computer and Electrical Engineering, Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on*, page 275, 2009.

[55] Family Health International, N. Mack, C. Woodsong, and United States. Agency for International Development. *Qualitative Research Methods: A Data Collector's Field Guide*. FLI, 2005.

[56] Herbert J Rubin and Irene S Rubin. *Qualitative interviewing: The art of hearing data*. Sage, 2011.

[57] Heidi Hardt. *NATO's Lessons in Crisis: Institutional Memory in International Organizations*. Oxford University Press, 2018.

[58] Zengyang Li, Paris Avgeriou, and Peng Liang. A systematic mapping study on technical debt and its management. *Journal of Systems and Software*, 101:193 – 220, 2015.

[59] Aubrey L. Mendelow. Stakeholder mapping. *Proceedings of the 2nd International Conference on Information Systems, Cambridge, MA*, 1991.

[60] Per Runeson, Martin Höst, Rainer Austen, and Björn Regnell. *Case Study Research in Software Engineering – Guidelines and Examples*. John Wiley & Sons, 2012.

[61] ISO/IEC/IEEE international standard - systems and software engineering – software life cycle processes. *IEEE STD 12207-2008*, pages 1–138, Jan 2008.

[62] Object Management Group. *Business Process Model and Notation (BPMN), Version 2.0*. January 2011.

[63] Philippe Kruchten. *The rational unified process : an introduction.* Addison-Wesley object technology series. Reading, Mass. : Addison-Wesley, 2000, 2000.

[64] Daniel Moody. The "physics" of notations: Toward a scientific basis for constructing visual notations in software engineering. *IEEE Transactions on Software Engineering*, 35(6):756–779, Nov 2009.

[65] Elias Gabrielsson and Christoffer Lundgren. ECPM Process Map - Combitech Case. `https://ipfs.gabson.se/ipfs/QmYS74GpASCrsuq1GXweUzkzTdred874aNRkcY7ezknPKZ`, 2018.

Appendices

Appendix A
ECPM Interview Guide

This appendix presents the interview guide that was used to elicit information, which later formed the ECPM Process Map, using formal interviews with the staff at Combitech.

Background

Q1: What is your role in the organisation? Is there a clear name for your role? Is there a role description? If you have several roles, please state how much of your time that you spend on each role.

Q2: How long have you been working at Combitech?

Q3: How long have you been working within the software industry?

Q4: What is your educational background?

Q5: Which technical support systems and platforms do you use when you are working?

Process Elicitation

Q6: What activities do you participate in?

Q7: Output: Where do you send the results of your work? In what form are the results sent? How are the results saved? What does the handover look like?

Q8: Input: What do you need to start a specific activity?

Q9: Control: Which guidelines do you follow? Can you work in any way you want?

Q10: Support: What do you use for support when you are working? Any specific colleagues or tools?

Information Flows

Q11: Make a stakeholder mapping together with the interviewee.

Q12: Who are you working with on a daily basis?

Q13: Which meetings, phone calls, e-mails, etc. are required to solve your tasks?

Q14: Who approves the output you produce in various activities?

Q15: Who or what controls that the purpose of the process you are working with is fulfilled? What happens if it is not fulfilled?

Q16: How much of the information that you receive is structured and written, i.e. "solid"? What type of information is this?

Q17: How much of the information that you receive is unstructured and verbal, i.e. "fluid"? What type of information is this?

Q18: How much of the information that you receive come through your manager and how much come directly from the source?

Other

Q19: Who do you consider should be the next interviewee?

Appendix B

Evaluation Interview Guide

This appendix presents the interview guide that was used to evaluate the process map, and thereby the ECPM Method and Notation Language, by eliciting information from the staff at Combitech using formal interviews.

Q1: What purposes do you see for the ECPM map? For example, what problems do you think it may solve?

Q2: Do you consider a value-based perspective to be applicable and relevant for your role?
Please rate '1' to '4', where '1' is equivalent to "Not Applicable or Not Relevant" and '4' is equivalent to "Highly Applicable or Highly Relevant".

Q3: Do you experience the ECPM Process Map to be supportive for your work?
Please rate '1' to '4', where '1' is equivalent to "Not Supportive" and '4' is equivalent to "Highly Supportive".

Q4: Do you think that the level of detail is well chosen?
Please rate '1' to '4', where '1' is equivalent to "Too detailed or too vague" and '4' is equivalent to "Well balanced".

Q5: Do you experience the ECPM Process Map to improve your understanding for your work in the big picture?
Please rate '1' to '4', where '1' is equivalent to "Does not improve my understanding at all" and '4' is equivalent to "Does improve my understanding greatly".

Q6: How well does the ECPM Process Map help you explain your work for your colleagues?
Please rate '1' to '4', where '1' is equivalent to "Does not help me explain at all" and '4' is equivalent to "Does help me explain very well".

Q7: How complete in terms of recognised processes do you experience the ECPM Process Map to be?

Please rate '1' to '4', where '1' is equivalent to "Lacks many processes" and '4' is equivalent to "Contains most or all processes".

Q8: Do you experience that the input and output objects between processes are generally correct?
Please rate '1' to '4', where '1' is equivalent to "Many errors" and '4' is equivalent to "Mostly correct".

Q9: Does the ECPM process map illustrate how your work is controlled and lead?
Please rate '1' to '4', where '1' is equivalent to "Not visualised" and '4' is equivalent to "Visualised clearly".

Q10: Do you experience the roles on the ECPM Process Map to be correct?
Please rate '1' to '4', where '1' is equivalent to "Many roles missing or wrong" and '4' is equivalent to "Most or all roles correct".

Q11: Do you think that it would be easy to reconfigure or update the ECPM Process Map (e.g. add or remove processes, output artifacts, roles, etc.) in order to describe new projects?
Please rate '1' to '4', where '1' is equivalent to "Very hard" and '4' is equivalent to "Very easy".

Q12: How much time do you estimate one would need to grasp the content of the ECPM Process Map?

Q13: Has this research project taken any of your time, apart from the activities during which the researchers were present? If so, how much?

EXAMENSARBETE A Method for Value-Based Process Modelling in Software Engineering
STUDENT Christoffer Lundgren, Elias Gabrielsson
HANDLEDARE Elizabeth Bjarnason (LTH)
EXAMINATOR Per Runeson (LTH)

Processmodellering inom mjukvaruindustrin

POPULÄRVETENSKAPLIG SAMMANFATTNING **Christoffer Lundgren, Elias Gabrielsson**

Processorganisation används av företag för att förbättra ledning och uppföljning av arbete. Detta examensarbete tog fram en metod för att skapa processkartor som ger en översikt över en organisations mjukvaruutveckling genom att beskriva dess värdeförädlingen med hjälp av processer.

För att kunna leda och utveckla ett företag eller annan organisation måste man som företagsledare kunna skaffa sig en överblick över verksamheten. Vidare behöver man kunna mäta olika egenskaper i organisationen för att kunna förbättra den. Även anställda som ej leder verksamehten kan lättare förstå sin roll genom att kunna se sig själva i det större sammanhanget.

I vårt examensarbete har vi utvecklat en metod och en samling grafiska symboler som när de tillämpas tillsammans skapar en översikt över en organisation.

Metoden beskriver hur man genom intervjuer och seminarier inhämtar information från företaget, för att sedan analysera den. De grafiska symbolerna används för att rita ut den analyserade informationen som en karta, vilken blir flera meter lång, där olika arbetsuppgifter är utritade.

Kartan blir ett kraftfullt verktyg för att t.ex. överblicka arbetsuppgifter, förstå hur mjukvara utvecklas, hitta aktiviter som är överflödiga och kan tas bort och därigenom sänka företagets kostnader, underlätta presentationer av hur företaget är organiserat för nyanställda, kunder eller andra intressenter, med mera.

Vi har valt att införa begreppet *värde* i kartan som en gemensam nämnare för alla aktiviteter som ett företag ägnar sig åt. Nära relaterat till värdebegreppet är termen *process*. En process tar en *input*, och *förädlar* den på något vis, och matar ut den som en *output* med högre värde. Vad som skickas in och ut kan både vara väldigt konkret, t.ex. järnmalm som förädlas till stålplåt, eller abstrakt som att mjukvara säkerställs inneha en viss funktionalitet med hjälp av testning.

Vi applicerade vår metod på Combitech AB och visade där hur man kan presentera en kombination av klassiska vattenfallsmodeller och moderna agila modeller i samma processkarta. Därigenom kan t.ex. en enhetschef, som betraktar organisationen på ett mer traditionellt vis, och mjuvaruutvecklare, som styrs av agila metoder, lättare förstå varandra.