

UNIVERSITY OF LJUBLJANA  
FACULTY OF ECONOMICS

MASTER'S THESIS

**APPLICATION OF BUSINESS PROCESS MANAGEMENT IN GROWING  
SOFTWARE DEVELOPING COMPANY INCLUDE IT GmbH**

Ljubljana, May 2016

ALEKSANDAR MARKELIĆ

## AUTHORSHIP STATEMENT

The undersigned Aleksandar Markelić, a student at the University of Ljubljana, Faculty of Economics, (hereafter: FELU), declare that I am the author of the master thesis entitled Application of Business Process Management in Growing Software Developing Company Include IT GmbH, written under supervision of red. prof. dr. Mojca Indihar Štemberger.

In accordance with the Copyright and Related Rights Act (Official Gazette of the Republic of Slovenia, Nr. 21/1995 with changes and amendments) I allow the text of my master thesis to be published on the FELU website.

I further declare

- the text of my master thesis to be based on the results of my own research;
- the text of my master thesis to be language-edited and technically in adherence with the FELU's Technical Guidelines for Written Works which means that I
  - cited and / or quoted works and opinions of other authors in my master thesis in accordance with the FELU's Technical Guidelines for Written Works and
  - obtained (and referred to in my master thesis) all the necessary permits to use the works of other authors which are entirely (in written or graphical form) used in my text;
- to be aware of the fact that plagiarism (in written or graphical form) is a criminal offence and can be prosecuted in accordance with the Criminal Code (Official Gazette of the Republic of Slovenia, Nr. 55/2008 with changes and amendments);
- to be aware of the consequences a proven plagiarism charge based on the submitted master thesis could have for my status at the FELU in accordance with the relevant FELU Rules on Master's Thesis.

Ljubljana, \_\_\_\_\_  
(Month in

Author's signature: \_\_\_\_\_  
words / Day /

Year,

## TABLE OF CONTENTS

<b>INTRODUCTION</b> .....	<b>1</b>
<b>1 BUSINESS PROCESS MANAGEMENT</b> .....	<b>2</b>
1.1 BPM Lifecycle – Business-Organizational View .....	4
1.2 BPM – Technical-Engineering View .....	6
1.3 BPM For Software Development .....	10
1.3.1 Agile.....	10
1.3.2 Capability Maturity Model.....	11
1.3.3 Key Performance Indicators.....	13
1.4 Success and Failure Factors .....	15
<b>2 BONITA BPM 7</b> .....	<b>17</b>
2.1 Bonita Studio.....	19
2.2 Bonita Portal.....	19
2.3 Bonita BPM Engine.....	22
<b>3 INCLUDE IT GmbH</b> .....	<b>23</b>
3.1 Introduction and History of Include IT GmbH .....	23
3.2 AgenaTrader .....	25
3.3 AgenaTrader – AdminTool .....	26
<b>4 BPM LIFECYCLE – INCLUDE IT GmbH</b> .....	<b>27</b>
4.1 Process Discovery and Analysis.....	27
4.1.1 OTRS .....	28
4.1.2 RedMine.....	29
4.1.3 Forums, Emails and Phone Calls .....	32
4.1.4 Software Release.....	32
4.2 Process Redesign.....	33
4.2.1 RedMine subproject “BetaTesters” .....	33
4.2.2 New Support Contact Form .....	35
4.2.3 AgenaTrader Wiki Project .....	36
4.2.4 AgenaTrader Education .....	36
4.2.5 Key Performance Indicator implementation .....	37
4.2.6 Build With Tags .....	38
4.2.7 Description of Reported Bugs.....	39
4.2.8 Software Release Cycle .....	40
4.2.9 Bug Fixing/Feature Developing.....	41
4.3 Process Implementation .....	43
4.4 Process Monitoring and Controlling .....	46
<b>5 PROPOSALS FOR FURTHER IMPROVEMENTS</b> .....	<b>47</b>
<b>CONCLUSION</b> .....	<b>51</b>
<b>REFERENCE LIST</b> .....	<b>56</b>

## LIST OF FIGURES

Figure 1. BPM Evolution .....	2
Figure 2. BPM Life-Cycle.....	4

Figure 3. The Six Core Elements of BPM .....	7
Figure 4. Rural and Urban Internet Penetration Rates in China, 2008 - 2015 .....	8
Figure 5. IBM Business Process Manager Mobile Application.....	9
Figure 6. BPR Success Factors .....	16
Figure 7. BPR Failure Factors.....	17
Figure 8. Starting Screen of Bonita BPM 7.0.2 .....	18
Figure 9. Google Trends - Searched Term “bonita bpm” .....	18
Figure 10. Bonita Studio .....	19
Figure 11. Bonita Portal Login Screen.....	20
Figure 12. Bonita Portal .....	20
Figure 13. Travel Request Form, Helen’s Side.....	21
Figure 14. Travel Request Form, William’s Side .....	21
Figure 15. Bonita BPM Engine.....	22
Figure 16. Include IT GmbH, Hierarchical Organization .....	24
Figure 17. AgenaTrader Andromeda 1.8 .....	25
Figure 18. OTRS .....	28
Figure 19. OTRS AS-IS Model “Dealing with Customer’s Request” .....	29
Figure 20. RedMine .....	30
Figure 21. RedMine AS-IS Model “Project Management” .....	31
Figure 22. Forum, Emails and Phone Calls, AS IS Model .....	32
Figure 23. Software Release Cycle, AS IS Model .....	33
Figure 24. Example of Release Note on BetaTester Project.....	34
Figure 25. Support Contact Form .....	35
Figure 26. AgenaTrader Wiki Project.....	36
Figure 27. Mercurial .....	38
Figure 28. Example of Additional Description in Redmine Issue .....	39
Figure 29. Software Release Cycle, TO BE Model .....	41
Figure 30. TO BE Model of Bug Fixing and New Feature Developing .....	42
Figure 31. Auto Build Email.....	44
Figure 32. Additional Warning for Support Contact Form.....	44
Figure 33. RedMine KPI / Burndown Chart .....	45
Figure 34. Assigning Priority Guideline .....	49
Figure 35. Redmine 1.8 Issues per Months.....	50
Figure 36. Number of OTRS Cases Opened per Month .....	52
Figure 37. Number of Unhandled Exception Cases Opened per Month .....	53

## LIST OF TABLES

Table 1. Internet of Things Units Installed Base by Category (billions), 2013-2020.....	3
Table 2. Process Analyst vs Domain Expert.....	4
Table 3. Process Areas, Categories, and Maturity Levels in CMMI 1.2 .....	12
Table 4. RedMine KPI / Bottlenecks (14 December 2015) .....	46

Table 5. Bottlenecks with All Priorities, (15 December 2015) .....47

Table 6. Bottlenecks with Only Low Priorities, (15 December 2015).....48

Table 7. Bottlenecks with Only Immediate Priorities, (15 December 2015).....48



## INTRODUCTION

In today's highly competitive and globalized economy, every company, in order to survive and progress, must use all available opportunities for its benefits. One of those opportunities lies in Business Process Management (hereinafter: BPM). Since observed subject is software developing company, application of BPM would be slightly different then "usual". In production or trading companies we have product(s) which is not so much abstract as thousands lines of code. Implementing new features, releasing new versions, debugging, testing are just some of processes which exist in this enterprise and yet they are not so much mentioned in classical literature regarding Business Process Management. Some of those processes have potential to be automatized and by doing so, a lot of working hours could be saved but, automation without efficiency leads just towards increased inefficiency. For that reason, before any involvement of a new information technology (hereinafter: IT) infrastructure, those processes would need to go through thoroughly optimization and maybe even Business Process Reengineering (hereinafter: BPR). Purpose of my work would result in my contribution to the body of knowledge about BPM in growing companies in IT industry. At the time of writing this thesis, unproportionally small number of scientific works was published regarding this field - for that same reason I have opinion that this master thesis could be scientifically valuable for BPM community. As already mentioned, observed object is company Include IT GmbH and its processes. Complete work regarding Business Process Model and Notation (hereinafter: BPMN) is planned to be done with open-source software Bonita BPM 7.

In consideration off all above written, there are three main objectives of this master thesis:

- optimizing existing processes and designing new ones for maximizing end customer satisfaction;
- define responsibility for each event in developing process;
- test capabilities of Bonita BPM 7 with software developing companies.

Structure of the thesis would consist five major fields:

- first chapter as an intro, would deal with theory literature review regarding business process management;
- second chapter would deal with BPMN software BONITA BPM 7 and its components;
- in the third chapter observed object would be described (Include IT GmbH) as well its two main products – AgenaTrader Andromeda and AgenaTrader AdminTool;
- at the fourth chapter, full BPM life-cycle would be conducted, from defining the current model of processes, model a new desired state, execute its implementation, monitor its performance and at the end do additional optimizing and fine tuning of newly implemented processes;
- last chapter would be dedicated to a final results of this work, which would be in form of interviews, and surveys and personal experience.

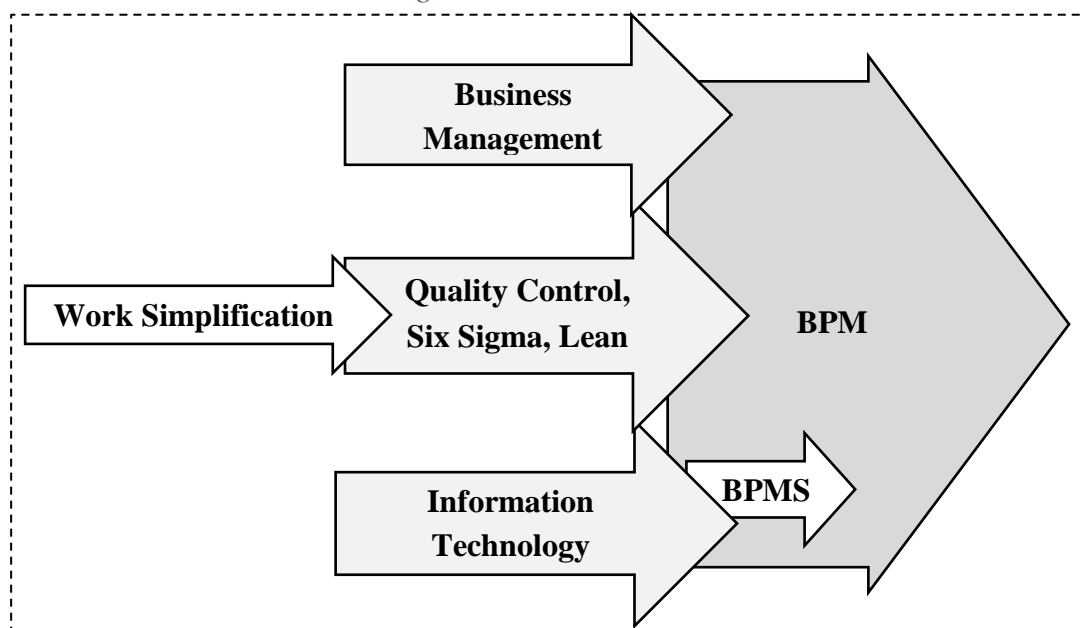
Reconstruction of IT Include GmbH processes could do a lot for this young enterprise. Since this is company which is constantly growing, each step forward regarding business processes is “easier today than tomorrow”, because each new employee brings new challenges for initial changes. If BPM optimization is done in this early stage, that would pave the road for future stable expansion. Expected results are improved customer satisfaction, structured quality control of final code which should result in more stable executions and decreased workload of employees by eliminating “double work” – just as Hammer once said “Don't Automate, Obliterate” ( Hammer, 1990, p. 104)

## 1 BUSINESS PROCESS MANAGEMENT

BPM is in same time art and the science (Dumas et al. 2013, p. 1). Since it is a relatively young science there are multiple definitions of BPM:

- “... is a field in operations management that focuses on improving corporate performance by managing and optimizing a company's business processes”( Harmon, 2005, p. 1).
- “... is the discipline of managing processes (rather than tasks) as the means for improving business performance outcomes and operational agility. Processes span organizational boundaries, linking together people, information flows, systems and other assets to create and deliver value to customers and constituents” ( Business Process Management - BPM - Gartner IT Glossary, 2015).
- “...is supporting business processes using methods, techniques, and software to design, enact, control, and analyze operational processes involving humans, organizations, applications, documents and other sources of information (Aalst et al., 2003, p. 4).

*Figure 1. BPM Evolution*



Source: P. Harmon, *The Scope and Evolution of Business Process Management*, 2010, p. 38.



One of the main theories regarding BPM and its evolution was devised by Harmon ( 2010, p. 38) According to him, BPM is clear synthesis from Business Management, quantitative methods like Six Sigma, Lean and other in combination with Information Technology (Figure 1). This shows that, Business Process Management is in middle of computer science (technical-engineering approach) and particular set of people's skills which is required for conducting business (business-organizational approach). Very often goal(s) of BPM cannot be achieved if person who is leading project do not have necessary people's skills which enables him to implement project successfully and improve business processes. Term "improvement of business process" means that its costs are reduced, execution time is decreased or just that error rate is brought to smaller or acceptable level (Dumas et al., 2013, p. 1).

Development of BPM is not finished, it is forming in accordance with current technology and trends. Nowadays we are witnessing exponential growth of devices connected to internet – so called Internet of Things (hereinafter: IoT). The best indication of dynamics of this phenomena can be seen in Table 1:

Table 1. Internet of Things Units Installed Base by Category (billions), 2013-2020

Category	2013	2014	2015	2020
Automotive	96,00	189,60	372,30	3.511,10
Consumer	1.842,10	2.244,50	2.874,90	13.172,50
Generic Business	395,20	479,40	623,90	5.158,60
Vertical Business	698,70	836,50	1.009,40	3.164,40
Grand Total	3.032,00	3.750,00	4.880,60	25.006,60

Source: *Gartner Says 4.9 Billion Connected "Things" Will Be in Use in 2015, 2014.*

Mentioned trend is starting whole new branch of Business Process Management - BPM Everywhere (hereinafter: BPME) ( Nathaniel, 2015, p. 15). Nevertheless, increasing number of IoT doesn't mean anything by itself – meaningful cooperation between them, and ability to deliver required final result does. Future brings whole new era in BPM, where IoT could be used as sensors and providers of data while old fashioned system would process data (data mining) and up on that, bring decisions and move processes forward. Data collection do not end with just simple sensors but very often it capture social networks as well ( Aalst & Song, 2004, p. 3). Tweets, social network statuses are analyzed and processed via semantic tools, converting qualitative data into quantitative and making those suitable as input information for advanced systems of decision.

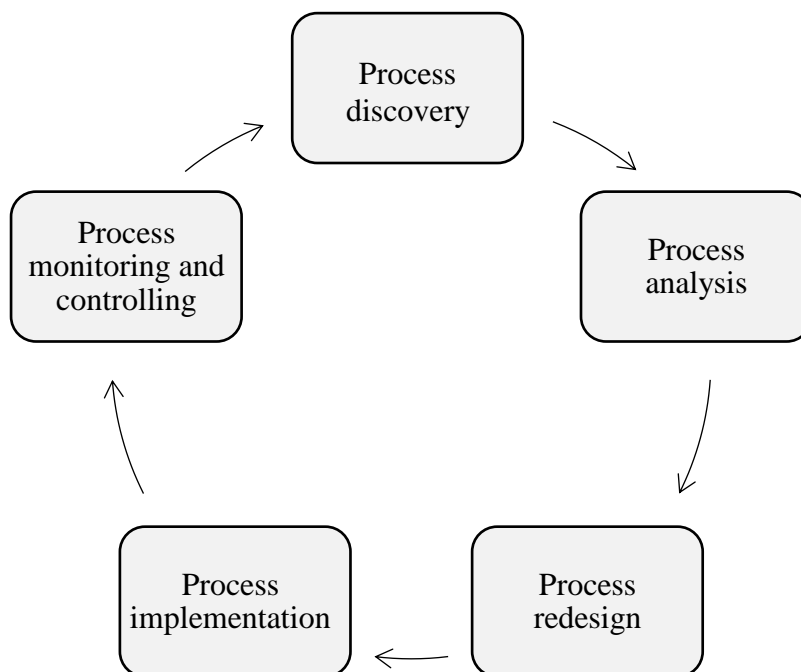
Since BPM is still young branch of science, depending from different authors we are having different view as well – especially regarding business-organizational or technological-engineering approach. Those two views are not in confrontation whatsoever but only putting emphasize on different elements of BPM

## 1.1 BPM Lifecycle – Business-Organizational View

Business Process Management in its methodology should always follow certain set of activities in proper order, as it can be seen in Figure 2 – this continuous workflow is called BPM Lifecycle:

- **Process discovery:** Starting point of all BPM auditing. During this activity responsible person (process analyst) should design AS IS process model which would truly represent situation in observed subject. Process model are done in correspondence with Business Process Model and Notation methodology. Responsible person should gather all data regarding complete business workflow, employees and their activities.

Figure 2. BPM Life-Cycle



Source: Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A, *Fundamentals of Business Process Management*, 2013, p. 21.

- **Process analyzing.** During process discovery, all possible metrics about collected processes are collected as well. This is done so in the end we could compare performance, before and after BPM. Here BPM theory is offering two possibilities: process to be done by process analyst or domain expert. This is especially important to emphasize when we are dealing with process that are highly specialized. Then we are facing with a two choices like shown in Table 2:

Table 2. Process Analyst vs Domain Expert

Aspect	Process Analyst	Domain Expert
Modeling Skills	Strong	Limited
Process Knowledge	Limited	Strong

There is no clear answer what should be done in this phase, it is always depending from current situation and industry. If company which is audited belong to industry of high specialization, then it is better that this process is done by domain expert. Otherwise this part could be done by process analyst.

- **Process redesigning:** During redesigning best practices should be applied to AS IS model – making them TO BE models. Firstly removing processes that are redundant and automate other where budget and infrastructure allows it. At this stage process analyst is still dealing with just models. Before any further step, towards implementation, those new processes in model can be tested in virtual environment and decision up its implementation can be given while reviewing its results.
- **Process implementation:** After modeling is done, process analyst have to make steps towards hardest and most critical part – implementation or execution phase. As previously mentioned this could mean many changes for employees. From losing a job, changing work position, to changing work habits, company culture or just reducing work load by using more productive technology. But BPM is not only about workers, it is customer oriented approach – that means that final result should not be measured by perception of workers but primarily by its output in form of increased/decreased customer satisfaction. Best KPIs (key performance indicators) for this abstract category are sales, net profit and customer satisfaction measured by their feedback.
- **Process monitoring and controlling:** When implementation stage is successfully finished, that means that each process and final output needs to be constantly monitored. For this stage proper Key Performance Indicators (hereinafter: KPIs) should be used in order to track overall performance. When environment and nature of project allows it, Business Activity Monitoring (hereinafter: BAM) software can be used. Currently mostly often this software solution are being used by financial institutions after BPM implementation. Using KPIs or BAM software as guide line, we need to recognize bottlenecks in workflow and eliminate or at least reduce them to acceptable level. This is never ending process since every company, just as its markets, have dynamical nature.

Named activities represent continuous efforts for improvement of business processes. They move in cycles because internal and external (buyers and suppliers) environment changes as well as technology.

During previously explained BPM auditing process, various stakeholders are taking place in BPM lifecycle:

- **Management team.** Most important segment of BPM Lifecycle. This is usually CEO or any other member of Top Management hierarchy. Management team should be driving force

behind whole process – they should support and enforce all necessary changes for project's sake.

- **Process owner.** Employee which is responsible for observed process. Under his responsibility he can have multiple employees for which work/results he is responsible or he can run one process all by himself.
- **Process participant.** All employees which are working day to day basis and taking part in examined business process. Their activities are often coordinated by process owner which is responsible for their results.
- **Process analyst.** In most cases person from external company, a consultant which is working on process discovery, analysis, redesign and implementation. It is highly recommended that this person have dual background – IT and business where he is conducting consulting/auditing. In order to be successful, all his activities must be coordinated and backed up by management team.
- **System engineer.** In accordance with process analyst specification, system engineer is developing suitable IT solution. His background and tasks are strictly technological.

In accordance with represented methodology, BPM audit would be applied later, in practical case on Include IT GmbH.

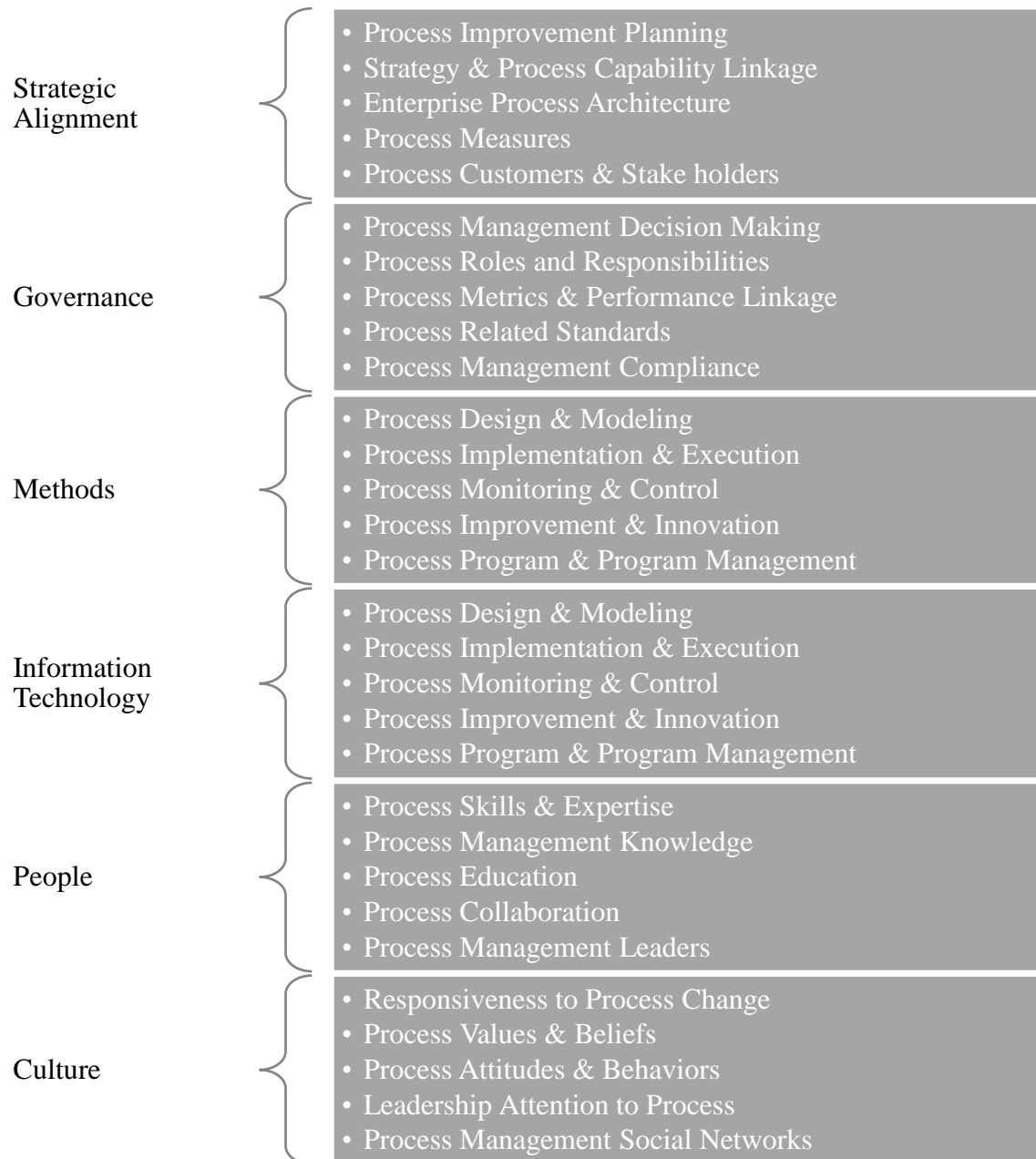
## 1.2 BPM – Technical-Engineering View

Previously was mentioned that BPM have two main stands in theory perspective. One of those perspectives is giving insights in “core of BPM” (Figure 3). It is interesting that this view have wider perspective since it includes BPM lifecycle and adds additional categories as well. Short overview of this theory of Rosemann & Brocke ( 2010, p. 112) can be seen by dividing them in six categories:

- **Strategic alignment.** First and most important element of BPM. When subject is experiencing BPM transformation, it must have support “from above”. Top management with CEO included, must give total support for this process. Only with this kind approach changes are possible, especially processes which are running in same way for several years.
- **Governance.** When implementing new process structure, it is important that transparency plays main role. It means that it must be known why some process is changed, who is in charge of that and when those planned operation would take a place. When responsibility roles are divided, that gives possibility to instantly identify “the weakest link” in case of obstacles.
- **Methods.** BPM methods are included in BPM theory. They can vary from case to case, from Six Sigma approach to Capability Maturity Model (hereinafter: CMM) – it is all depending from responsible person who is in charge for BPM implementation. Decision would be brought in accordance with industry where change is taking place, budget, number of employee and other elements relevant for this decision.
- **Informational Technology.** During the 90's BPM was almost only about process optimization without too much automatization. Nowadays, IT is rapidly expanding and it

can be used for various business processes. Any process in company, starting from office desk to factory line which can be automated – it should be. Modern software solutions are making this task possible and cheaper by each day. Software for this purpose vary from KPI creating tools to complete Business Process Management Suite (hereinafter: BPMS).

*Figure 3. The Six Core Elements of BPM*



Source: M. Rosemann & J. Brocke, *The Six Core Elements of Business Process Management*, 2010, p.112, Fig. 2.

- **People.** BPM is under constant pressure to evolve because people's (employee's) capabilities changes over time. During the 90's company would need additional training for employees to send an email, while today it is considered as part of elementary knowledge. This could be considered as external factor which is advantage for BPM implementation. In some cases when new processes are made, they could require additional education,

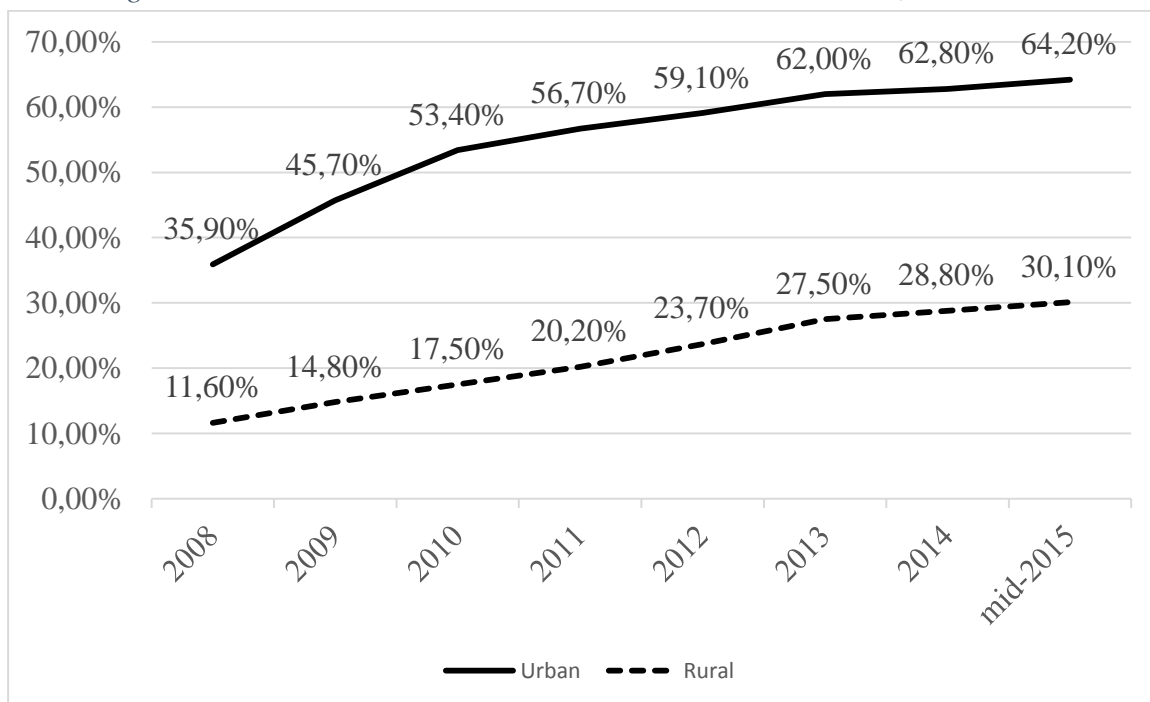
depending on their complexity and this could be challenging element for successful implantation.

- **Culture.** Corporate culture plays significant role in BPM. This is one of the hardest elements to change in process of BPM transformation and very often, element which would lead whole project to failure. Good example for that is 3M BPM case ( Harmon, 2010, p. 69), where Six Sigma was introduced and with it, new corporate culture – product oriented instead of previous innovation oriented. New method show increment of financial performance and in same time, reduction of sales of newly innovated products. As soon as CEO which implemented new methods moved to another company, 3M corporate culture eventually reverted to old state – all previous changes were canceled due to previous dominant corporate culture. Nature of corporate culture also depends from industry sector – usually IT industry is more open for changes while others with longer tradition are not.

Development and usage of BPM solution is constantly growing due to next factors:

- Average employee is getting more skilled in IT, year by year. This trend is happening not only in highly developed countries but also in rest of less developed world. Proof of this statement we can see in case of China in Figure 4.

*Figure 4. Rural and Urban Internet Penetration Rates in China, 2008 - 2015*



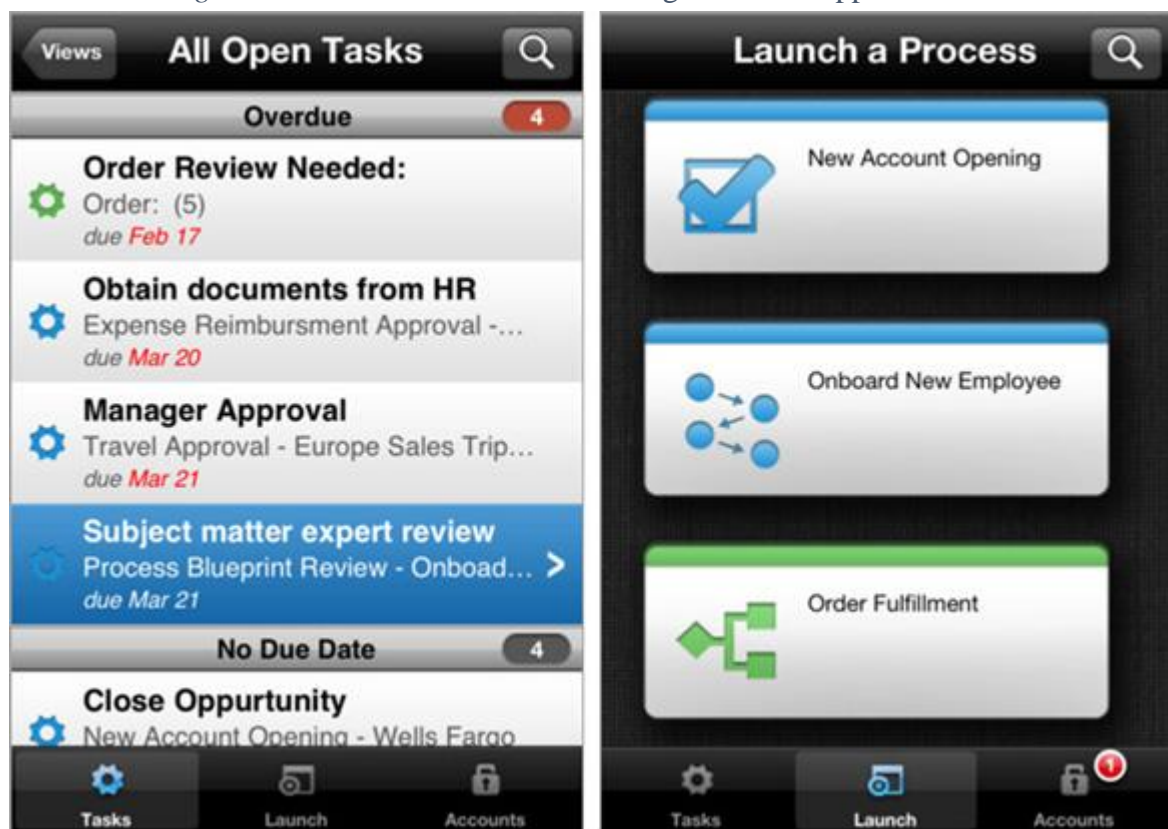
Source: *The Internet in China: How the Network is Shaping the PRC's Economic and Social Landscape*, 2016.

Named skills do not come from deliberate education provided by company or government, but more from everyday life where usage of smartphones, internet, computers and other sophisticated technologies is becoming just part of normal life. As final results there is direct

correlation between percentage of IT skilled workforce and chances to conduct successful BPM project and achieve higher productivity in audited company.

- Increasing number of BPM software on market. With single search on website Capterra it can be found 165 software solutions regarding BPM and many of those are even freeware ( Best Business Process Management Software | 2016 Reviews of the Most Popular Systems, 2016). This makes BPM solution available for each enterprise which is interested in their self-improvement.
- Availability of BPM suites as mobile apps. As previously said, using SOA and similar technologies BPM processes can be run on multiple platforms, including smartphones platforms like Android, iOS and Windows ( What's new in IBM Business Process Manager V8, 2016). By using mobile platforms employees are able to take part of all processes even when they are not at office desk – doing so information from “field” are available in database even before employee returns to his desk. This is small, but yet significant catalyst in future positive trends of BPM usage.

*Figure 5. IBM Business Process Manager Mobile Application*



*Source: What's new in IBM Business Process Manager V8, 2016.*

- Globalization and free market competition is one of the big drivers of BPM. Best example is again, Ford's case (Dumas et al, 2013, p. 11). Ford would never start to question their processes if they did not see that for approximately same job Mazda was using 16 workers instead Ford's 500. For example can be used European Union as a huge free market union, where market competition forces are even stronger today. Having this environment in force,

implementation of BPM could mean difference between successful enterprise or eventual bankruptcy.

## **1.3 BPM For Software Development**

When dealing with software developing companies, environment is more dynamical than in other industries and for that reason it requires additional solutions and methodology. One of those solutions is Agile ( What is Agile Software Development?, 2015)

### **1.3.1 Agile**

Non-profit organization Agile Alliance is actively promoting this principles through Agile Manifesto. Agile Manifesto was made by seventeen independent authors where they manage to find common principles regarding software development. In this document they described roadmap for introducing Agile which consist from twelve principles ( Principles behind the Agile Manifesto, 2001):

- Customer satisfaction by early and continuous delivery of useful software.
- Welcome changing requirements, even late in development.
- Working software is delivered frequently (weeks rather than months).
- Close, daily cooperation between business people and developers.
- Projects are built around motivated individuals, who should be trusted.
- Face-to-face conversation is the best form of communication (co-location).
- Working software is the principal measure of progress.
- Sustainable development, able to maintain a constant pace.
- Continuous attention to technical excellence and good design.
- Simplicity—the art of maximizing the amount of work not done—is essential.
- Self-organizing teams.
- Regular adaptation to changing circumstance.

In some cases certain contradictions can be found between Agile and BPM, but they are not critical obstacle for their coexistence. Agile as the name says for itself, represent not so much structured organization while BPM strive for opposite. This can be seen in Agile's statement of values ( What is Agile?, 2016):

1. Individuals & Interactions over Process & Tools.
2. Working Software over Comprehensive Documentation.
3. Customer Collaboration over Contract Negotiation.
4. Responding to Change over Following a Plan.

Most interesting point for this master thesis is the first one: “Individuals & Interactions over Process & Tools”. With this point, Agile Manifesto doesn't advocate that process and tools



should be ignored, but just that personal interaction in company is more important than pure process itself.

Second example would be face to face communication, which is not always possible by classical BPM framework. Very often processes run in form of web form which wait in queue to be solved/answered. Sometimes time needed for further response is several days and employees in this case could be located in completely different countries.

Another example of these differences can be seen if we observe BPM as a clear road to automation and Agile method of development named pair programming (Williams, 2001, p. 27). Named method requires two programmers are using one workstation – one would use mouse/keyboard and actively enter lines of code, while second would be there as observer, tracking code for possible mistakes and suggesting better solutions if one is possible. Even this method, from BPM standpoint, looks like enormous waste of time, it showed that final code results in 15% less bugs (Cockburn & Williams, 2000, p. 4). Decreased amount of bugs could save enormous amount of capital if final solution is used by either relatively high number of users or if developed software could cause high financial damage with developed bugs, all this could result in justification of double costs for programmer's salaries.

### 1.3.2 Capability Maturity Model

Very important BPM tool for software developing industry is Capability Maturity Model. CMM became famous during the 90s by US Defense of Department Request for CMM happened due to increasing number of delivered bugs in final version of deployed software and also, due to constant breaking deadlines and budget for final releases. For the named reasons, US Defense of Department asked Software Engineering Institute (hereinafter: SEI) at Carnegie Mellon University and Watts Humphrey to develop CMM (Paulk, 1993, p. 2).

Using US Defense of Department's data regarding organization that contracted for them, Humphrey and SEI made model which is able to put each software organization in one of five levels of maturity. By doing so, US Defense of Ministry had better information regarding reliability of their contractors. Of course, this screening impacted their decision regarding future projects.

- **Level 1 – Initial.** The lowest level of organization from CMM's perspective. Main characteristics of this initial stage are chaotic organizational environment and ad hoc processes. Success of this kind of organizations depends mostly up on efforts of few individuals. Whole system acts in manners of black box – with given inputs there is insecurity regarding what would outputs be.
- **Level 2 – Repeatable.** In order to advance from initial level to level two, CMM is not entirely clear - it is not giving instructions how do come there but it has clear idea what is actually level two. At this point certain existence of processes is obvious and their efforts to

achieve repeatable and constant results. First sign of project management can be established and KPIs are being used to define performance between processes and their results.

- **Level 3 – Defined.** Characteristic of this level of maturity is that all processes in this organization are defined which implies that achieving constant results is possible. This is done using project management and by documenting all software developing process. Software engineers as well other employees, completely understands their roles and responsibilities.
- **Level 4 – Managed.** At this stage KPIs are deeply integrated so managers can have information regarding progress and problems of each process in organization. This information gives them ability to objectively bring any further decision. From this point on, process capability is possible which implies in significant decrease in results variability.
- **Level 5 – Optimizing.** In last level, organization is optimizing existing processes by implementing new technologies, increasing productivity and keeping the step with possible organization expansion.

Table 3. Process Areas, Categories, and Maturity Levels in CMMI 1.2

Process Area	Abbreviation	Category	Maturity Level
Configuration Management	CM	Support	2
Measurement and Analysis	MA	Support	2
Process and Product Quality Assurance	PPQA	Support	2
Project Monitoring and Control	PMC	PM <sup>1</sup>	2
Project Planning	PP	PM	2
Requirements Management	REQM	PM	2
Supplier Agreement Management	SAM	PM	2
Decision Analysis and Resolution	DAR	Support	3
Integrated Project Management	IPM	PM	3
Organizational Process Definition	OPD	Process M <sup>2</sup>	3
Organizational Process Focus	OPF	Process M	3
Organizational Training	OT	Process M	3
Product Integration	PI	Engineering	3
Requirements Development	RD	Engineering	3
Risk Management	RSKM	PM	3
Technical Solution	TS	Engineering	3
Validation	VAL	Engineering	3
Verification	VER	Engineering	3
Organizational Process Performance	OPP	Process M	4
Quantitative Project Management	QPM	PM	4
Causal Analysis and Resolution	CAR	Support	5
Organizational Performance Management	OPM	Process M	5

<sup>1</sup> Project Management

<sup>2</sup> Process Management

CMM technique can be used as guide line while (re)making processes in software developing organization. Level 5 should be goal which should be eventually achieved.

Additional improvement of CMM methodology is done in Capability Maturity Model Integration 1.2<sup>3</sup> (hereinafter: CMMI 1.2). Methodology upgrade and detailed instructions regarding all needed processes in CMMI are done by Carnegie Mellon University in their work (CMMI Product Team, 2006, p. 5). CMMI processes are separated for three different areas:

- CMMI for Acquisition.
- CMMI for Development.
- CMMI for Services.

In Table 3 are listed all process areas for each maturity level of CMMI (in this master thesis, only CMMI for Development would be observed). CMMI went one step further that its predecessor CMM. For each process area, detailed list of all sub-processes are listed. For example, first step to maturity level 2 is to ensure that process area of Configuration Management (hereinafter: CM) is introduced CMMI Institute, 2010:

- CM 1. Baselines of identified work products are established.
  - CM 1.1 Identify configuration items, components, and related work products to be placed under configuration management.
  - CM 1.2 Establish and maintain a configuration management and change management system for controlling work products.
  - CM 1.3 Create or release baselines for internal use and for delivery to the customer.
- CM 2. Changes to the work products under configuration management are tracked and controlled.
  - CM 2.1 Track change requests for configuration items.
  - CM 2.2 Control changes to configuration items.
- CM 3. Integrity of baselines is established and maintained.
  - CM 3.1 Establish and maintain records describing configuration items.
  - CM 3.2 Perform configuration audits to maintain the integrity of configuration baselines.

Same is done for all the other process areas. In that way all maturity level are strictly defined and possibility for misunderstanding and wrong judgment is brought to its minimum.

### **1.3.3 Key Performance Indicators**

---

<sup>3</sup> Starting from 2011. CMMI 1.3 was introduced but it would not be examined in this master thesis

In order to manage processes and measure their efficiency, proper metrics are required. Implementing key process indicators in processes of software developing is quite complex task to be done. Production factory, for example, could easily measure workers productivity by counting produced units, malfunctions, used raw materials and similar. Same methodology and logic would not give satisfying results in software engineering. Best example for that is metric single lines of code (hereinafter: SLOC). Named methodology is counting new line of code to estimate programmer's productivity. Using SLOC as main KPI could be deadly for software developing company ( Rosenberg, 1997, p. 139). This is due to fact that all employees are working in accordance with system of rewards and sanctions, and if more lines of code would provide them more rewards then they would simply strive to do that – simple deliver more lines of code regardless of their quality. At the end of development, this code can be completely useless because quality of this code cannot be measured completely until it reaches final users or at least beta testers ( Bhatt, Tarey, & Patel, 2012, p. 151)

If another KPI would be used, for example, spent hours in development, company could have situation that people are just motivated to spend more time at their workplace, which does not imply that they would be more productive or valuable for the project. KPI can be set in Quality Assurance department by counting detected bugs in newly implemented code. But this could be false indicator, because amount of bugs would be heavily impacted by complexity of requested feature.

My personal view defers from named methodologies. Using purely quantitative metrics would not show real state of development. Since software engineering is combination of art and science ( Ruhe & Saliu, 2005, p. 48), it is very hard to apply pure formulas on its results. Automated software tests which seek mainly for unhandled exceptions, show only one part of software quality – its stability. While its usability is impossible to rate by another program – it is possible only by rating of its final users by manual testing ( Ramler & Wolfmaier, 2006, p. 88) at least this is the case with software that is developed by Include IT GmbH. For stated reasons, better choice for KPIs could be metrics from quality side like:

- Users ratings of software.
- Amount of sales.
- Filled complaints/bugs.
- Ratio of people which are using demo with full version users.
- Ratio of inflow of demo users/inflow of full version users.
- Users which stopped to use our software.

All named metrics in combination with a time units (per day, per month, etc.) can be used as KPIs of software engineering quality and productivity. In some cases SLOC could be used as metric like bugs per SLOC or similar. Some of those indicators are mutual results of marketing and sales efforts, but this kind of bias can be minimized if all metrics are observed together.

Overall productivity of one IT company can be observed and measured by using any project management tool in such way that we compare period to period state. This values can represent either number of features to implement or bug to fix – both can be used as forms of indicators. Good example for this case is online project management tool RedMine which would be further discussed in 4.1.2 chapter. All elements as features or bug fixes can be tracked and dynamics of their successful implementation can be compared from one period to another. This approach have disadvantage because not all elements have same complexity, thus different time (working hours) is needed for their implementation. Solution for this problem could be made in form of compromise, that each element is dedicated with specific value of working hours that is planned to be spent. This amount of time would be firstly suggested by developers and later should be approved (or disapproved) by controlling entity (head of development, project manager or CEO). Continuing from that point, having RedMine's issues represented as value of working hours, efficiency could be measured by checking how much issues is fixed in one week – for example, one developer which had 40 working hours at disposal, fixed/developed issues which required 10 working hours. Following this logic, this would mean that efficiency of observed developer is  $10/40 = 25\%$ . In this particular case, this would be sign for additional attention and investigation in order to understand reason for so low efficiency level. Reasons for that could be that wrong estimation regarding required hours was wrong from the start, or just developer took four times more time to develop/fix this issue due to his low skills or lack of motivation. Rate of tracking this KPIs is depending from company to company. If Agile methodology is being used, then one set of parameters should be used in same tempo which is designated as length of a one sprint (time period used in Agile methodology, during which specific work has to be completed and made ready for review.), while other set should be used day by day.

Final conclusion is that business should be devoted to its customers (first principle of Agile Manifesto) because they are source of incomes that gives life to a company and thus, one of the best indicator of software developing performance. User experience in combination with other metrics could result in quality KPIs which could be used for further improvements of business processes.

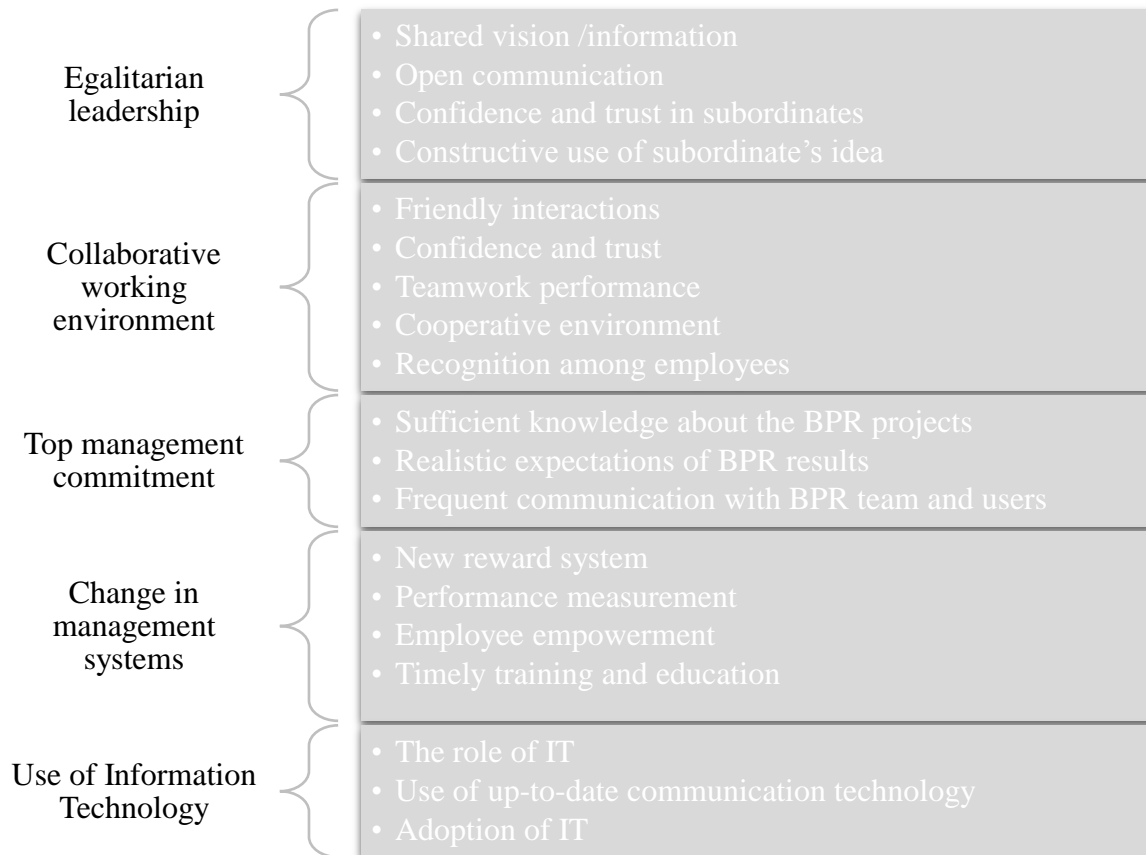
## **1.4 Success and Failure Factors**

In BPM/BPR industry there is well good known case during the 80's where company Ford managed to decrease number of their accounts payable ( Hammer, 1990, p. 107). From staggering number of 500 employees in that department, they managed to rise accuracy and speed while in same time, number of employees is decreased for 76% - even though, original goal was to decrease it for just 20%. In most BPM seminars this study case is used as an intro and intimidation factor.

But in very few cases we hear about all those unsuccessful projects. BPM is very complex and sometimes controversial idea for employees because very often it measures all workers performances and could lead to unwanted rearrangement of work positon and sometimes even

loss of job. It would be naive if we would believe that average worker would be happy because this changes are for “greater good”, for company’s sake, if that would mean that his service are not needed any more. In order to beat this kind of obstacles, BPM implementer need to have set of skills that are beyond those which can be learned at any business school. Without rich experience and full top management support, most of those projects are doomed – according to Ramesh only 20% of them are finished on planned time ( Your BPM Implementation is Bound to Fail | BPMInstitute.org, 2007)

*Figure 6. BPR Success Factors*



Source: M.N Habib , *Understanding Critical Success and Failure Factors of Business Process Reengineering*, 2013, p. 2.

Most essential success factors could be divided in several categories. Prof. Habib made list with self-explanatory categories and factors in Figure 6. In his paper, ( 2013, p. 2) prof. Habib explains in details all required action in order for successful BPR implementation – joint effort from top management to bottom level employee with help of latest IT. But we should bear in mind that top technology, infrastructure and technical knowledge means almost nothing in BPM without people’s skills which would should pave the road for other elements (Dumas et al, 2013, p. 23).

As counterpart to success factor, there are failure factors (Figure 7):

*Figure 7. BPR Failure Factors*



Source: M.N Habib, *Understanding Critical Success and Failure Factors of Business Process Reengineering*, 2013, p. 2.

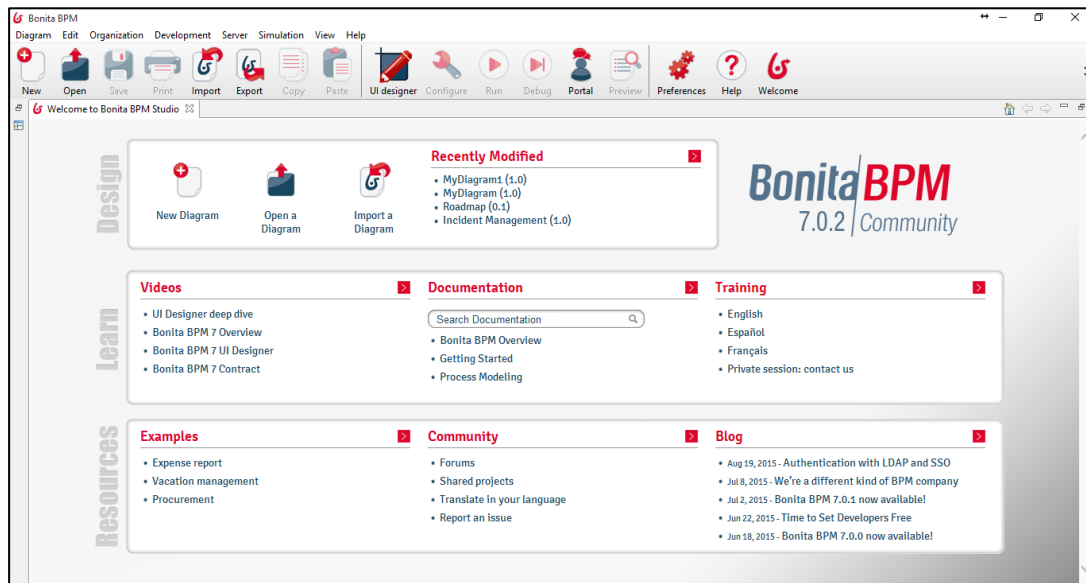
Each reorganization generates specific insecurities among employees. As it can be seen in Figure 7, that could be for various reasons. These situations occur when company face changes, employees, middle management and sometimes even top management are afraid for their position and income – for that reasons they would resist new implementations and try to retain “old state”. If they cannot be convinced otherwise – project outcome is questionable. In some cases, fear of losing job is justified like in case of company 3M where implementation of BPM principles required laying off 11% of workforce (8.000 people) ( Harmon, 2010, p. 69) or even more famous case, Ford’s acquisition of Mazda, where reduction in Ford’s payable department was 76%, from initial 500 to 120 employees (Dumas et al, 2013, p. 11).

It is up to top management and person in charge for BPM implementation to fight with named resistance factors ( Muehlen & Ting-Yi Ho, 2005, p. 463). They can be prevented (or at least reduced) by proper introduction, explanation and education regarding planned changed and new processes. As conclusion it could be said that final outcome depends from top management determination, skills of BPM implementer and available resources.

## **2 BONITA BPM 7**

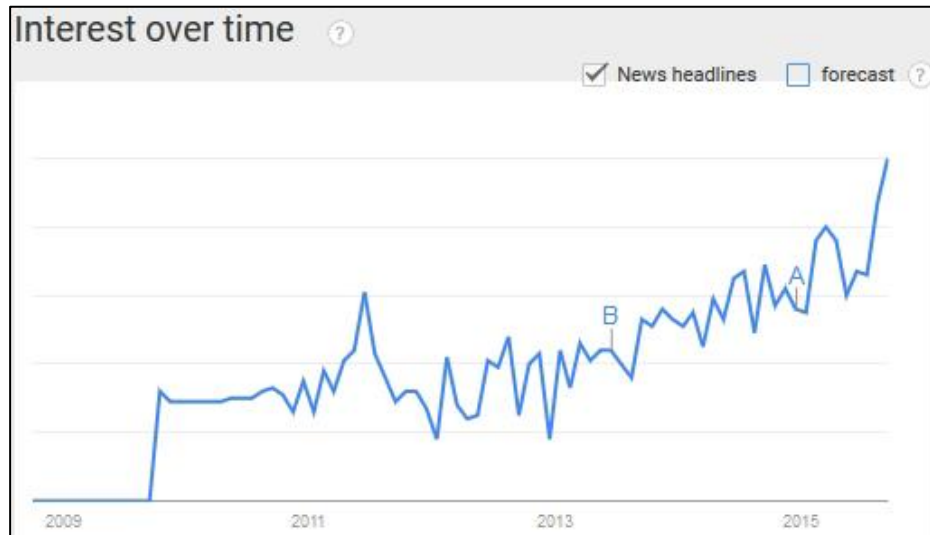
Bonita BPM (Figure 8) is open source suite created in France National Institute for Research in Computer Science in 2001. Starting from 2009, software Bonita is being developed by company Bonitasoft. Bonita was founded by Miguel Valdés-Faura and Charles Souillard – current CEO and Chief Technical Officer (hereinafter CTO) respectively. At the moment, they have over one thousand users and according to them, they are the fastest growing BPM open source community in the world. This positive trends attracted some big investors like Ventech, FSN PME, Auriga Partners, Serena Capital and others. Additional funds meant additional workforce which allowed them to increase number of developers.

Figure 8. Starting Screen of Bonita BPM 7.0.2



Judging by Google Trends (Figure 9) B is having constant growth of global interest. This is clear indication that Bonita BPM is having constant growth of popularity and it is probably at its first steps of potential popularity and that in the future it could set some standards in this industry

Figure 9. Google Trends - Searched Term “bonita bpm”



On their website they provided numerous video tutorials so new users would be able to adapt more easily. For any questions, user could choose from either official support team of Bonita or just search through forum in order to find solution. Beside all mentioned, users have at disposal webinars which give advance explanation for designing BPM solutions.

Infrastructure of Bonita BPM 7 can be divided on three main components:



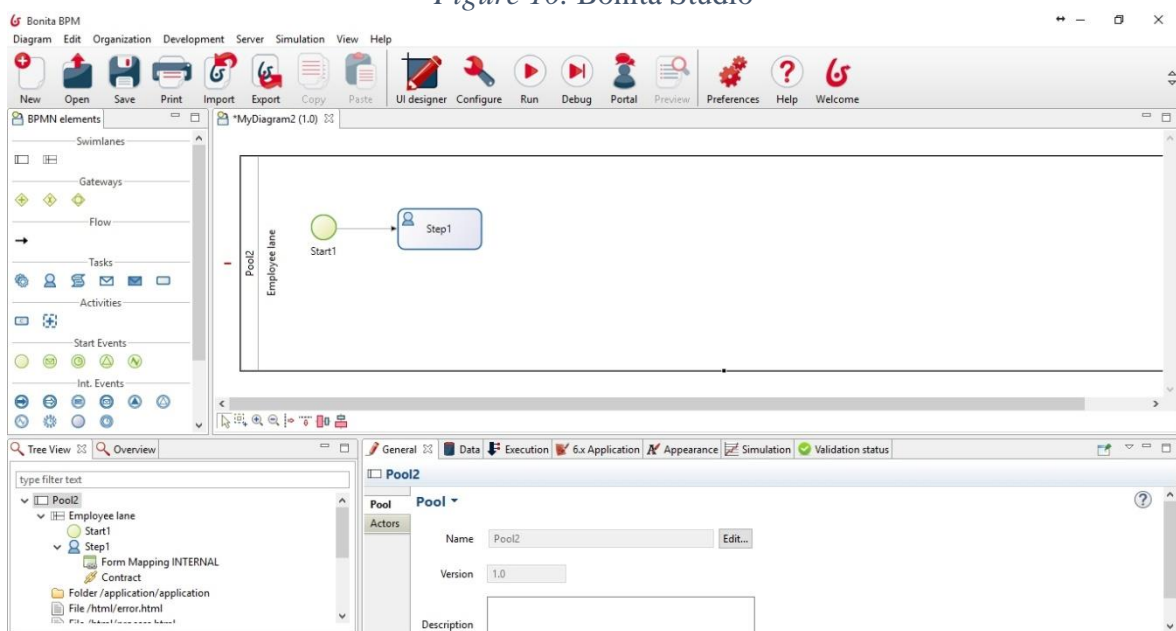
- Bonita Studio.
- Bonita Portal.
- Bonita BPM Engine.

## 2.1 Bonita Studio

Bonita Studio (Figure 10) is module where user actively design business processes with BPMN 2.0 (OMG, 2011, p. 1). It is divided in four main areas:

- Whiteboard, area where business processes are being designed together with pools – workflow models.
- BPMN palette, which consist all needed BPMN 2.0 elements for designing any business process.
- Tree view/overview, the panel with allow you to see the entire process diagram.
- Details panel, allows you to set and technical parameters regarding modified process. This panel turns workflow model into working application.

*Figure 10. Bonita Studio*



Bonita Studio is have user friendly interface which minimize need for any further training. If user have experience with BPMN standards, he would not need additional training for transition to this tool.

## 2.2 Bonita Portal

Bonita Portal module is running in web based form which is hosted on local HTTP server (Apache Tomcat / 7.0.55) – it can be run on local computer from URL: [http://localhost:8080/bonita/portal/homepage#?\\_p=tasklistingadmin&\\_pf=2](http://localhost:8080/bonita/portal/homepage#?_p=tasklistingadmin&_pf=2)

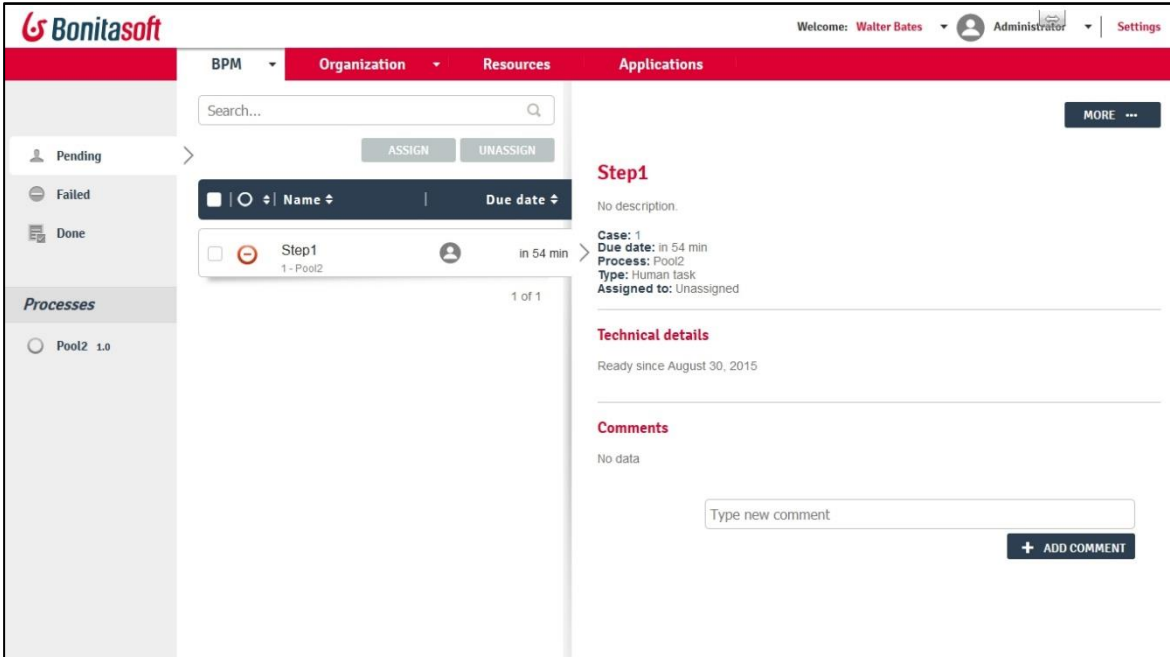
As soon as we try to activate any process Bonita would require from user to identify himself over Bonita Portal login form (Figure 11)

*Figure 11. Bonita Portal Login Screen*

The image shows the Bonita Portal login screen. On the left is the Bonitasoft logo, consisting of a red stylized '6' and the word 'Bonitasoft' in blue and red. On the right is the 'Login form' with two input fields: the first contains the username 'helen.kelly' and the second contains masked characters '\*\*\*'. Below the fields is a red 'Login' button with a white cursor icon pointing at it.

In accordance with set privileges and obligations specific Bonita Portal would appear in front of the user (Figure 12). From that point user can check his pending processes, failed one and one which are already done. Of course, he can start a new one – but in accordance with predefined rules which were already set in Bonita Studio. Basically all previously drawn and set BPMN models Bonita Studio, are set in force in Bonita Portal. It can be said that Bonita Portal is user's module while Bonita Studio is more BPM developer module.

*Figure 12. Bonita Portal*

The image shows the main interface of the Bonita Portal. At the top is a red header bar with the Bonitasoft logo on the left and 'Welcome: Walter Bates' followed by a user profile icon and 'Administrator' on the right. Below the header is a navigation bar with tabs: 'BPM' (selected), 'Organization', 'Resources', and 'Applications'. The main content area is divided into three sections. On the left is a sidebar with a search bar and a list of process states: 'Pending', 'Failed', and 'Done'. Below this is a 'Processes' section with a list item 'Pool2 1.0'. The middle section displays a table of tasks. The first task is 'Step1' with a due date of 'in 54 min'. The right section shows details for 'Step1', including 'No description', 'Case: 1', 'Due date: in 54 min', 'Process: Pool2', 'Type: Human task', and 'Assigned to: Unassigned'. Below this are sections for 'Technical details' (Ready since August 30, 2015) and 'Comments' (No data). At the bottom right is a text input field for 'Type new comment' and an 'ADD COMMENT' button.

One example of, how inside of one workflow same document can be seen in example of simple case of request from employee Helen Kelly for travel expenses (Figure 13).

*Figure 13. Travel Request Form, Helen's Side*

The screenshot shows the 'Travel Request for Helen Kelly' form in the Bonita BPM Portal. The form is titled 'See an application in the Bonita BPM Portal' and includes a navigation bar with 'Tasks', 'Cases', and 'Apps'. The form fields are as follows:

First Name Helen	Last Name Kelly
Email (change to your mail address to receive mails) idcbonita@gmail.com	Mobile 484-302-5187
Departing from: * Paris	Arriving at: * Bangalore
Departure Date * 06 Jul 2013	Return Date * 28 Jul 2013
Mode of Transport Flight	Travel Advance * 250
Hotel Booking Required? <input checked="" type="checkbox"/>	

A 'SUBMIT' button is located at the bottom center of the form.

Source: *See an application in the Bonita BPM Portal*, 2016.

After being submitted for approval, travel request form sent from Helen, arrives on queue of Helen's boss, William (Figure 14):

*Figure 14. Travel Request Form, William's Side*

The screenshot shows the 'Travel Request for Helen Kelly' form from William's perspective. The form is titled 'See an application in the Bonita BPM Portal' and includes a navigation bar with 'Tasks', 'Cases', and 'Apps'. The form fields are as follows:

First Name Helen	Last Name Kelly
Email idcbonita@gmail.com	Mobile 484-302-5187
From Paris	To Bangalore
Departure Date 06 Jul 2013	Return Date 28 Jul 2013
Mode of Transport Flight	Travel Advance * 250
Hotel Booking Required? <input checked="" type="checkbox"/>	

Below the form fields, there is an 'Approval Status' section with three radio buttons: 'Approved', 'Rejected', and 'Modify Request'. Below this is a 'Comments (in case of Modify Request)' text area. A 'SUBMIT' button is located at the bottom center of the form.

Source: *See an application in the Bonita BPM Portal*, 2016.

In case of William, he got complete travel request from Helen, with additional two fields: "Approval Status" and "Comments". In accordance with company's policy William then approve, reject or asks for modification where he would explain in Comments section. This was

small representation of simple use of Bonita, which could be applied to almost any BPM situation.

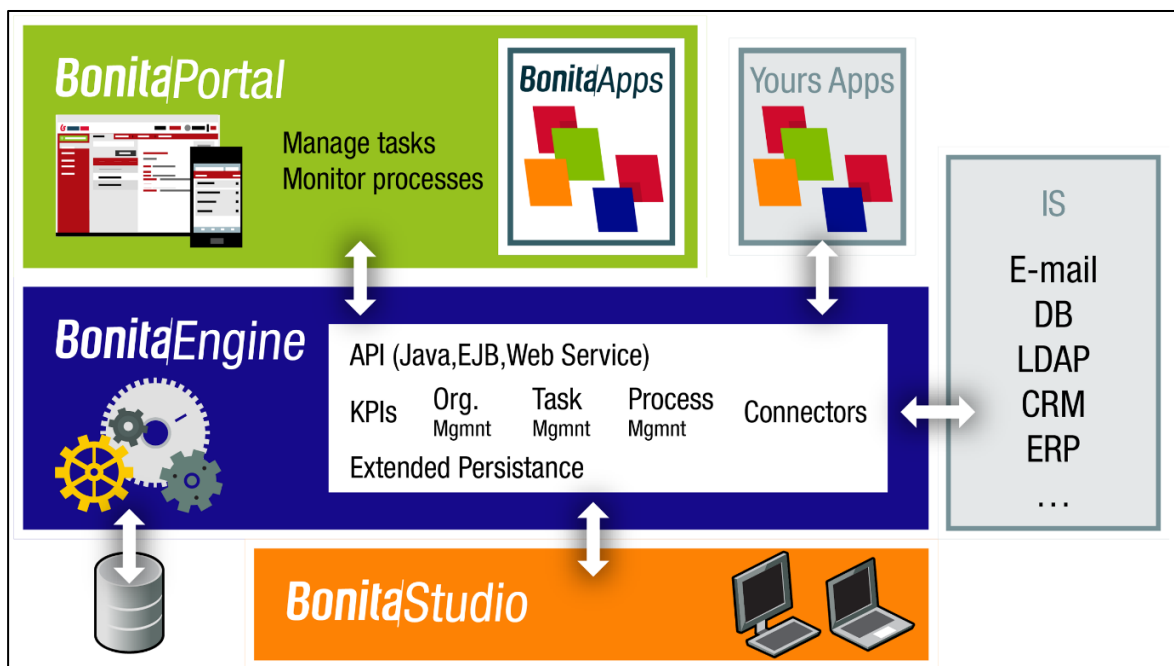
## 2.3 Bonita BPM Engine

Bonita BPM engine (Figure 15) is background module which is not visible for ordinary user – it uses various application programming interfaces (hereinafter: APIs) that allows us to connect Bonita's processes with other applications and web services.

On Figure 15 it explained how Bonita BPM engine is functioning as center of entire suite. All actions done in Bonita Studio are done with help of Bonita Engine and later are manageable with Bonita Portal or some other system with connection to API. Well-structured API technologies allows Bonita to successfully connect to other applications inside company.

In example of Include IT GmbH, this tool could be used to connect in same time with Open-source Ticket Request System (hereinafter: OTRS), RedMine and email server in same time. If deeper integration is needed, API could be extended to existing licenses database and by doing so complete informational system could be captured inside Bonita.

*Figure 15. Bonita BPM Engine*



*Source: Bonita BPM Engine architecture, 2015.*

Since Bonita Engine allows connection to customer relationship management (hereinafter: CRM) and enterprise resource planning (hereinafter: ERP) software, implementation is possible with some popular solutions like MS Dynamics, SAP ERP or some other. Constant updates of documentation regarding development of API and its free access, as a final results means easier

access for developers. This gives to Bonita whole new level of compatibility and usability for business users.

### **3 INCLUDE IT GmbH**

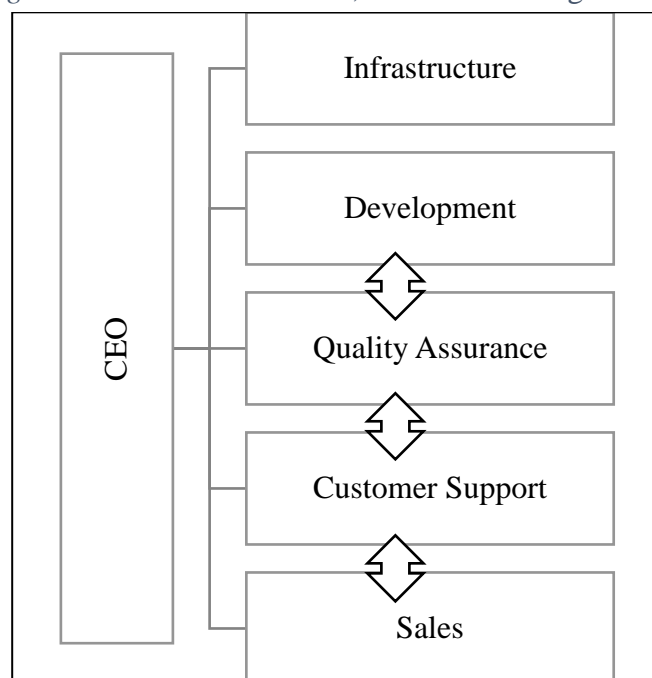
Observed subject for this master thesis is Austrian company, located in Vienna – Include IT GmbH. During researching and writing of this thesis, I was employee in this company so most of information regarding processes was collected directly from my department - Quality Assurance. Rest of qualitative data is collected by interviewing coworkers and CEO, while quantitative data was acquired by extracting csv files from RedMine and OTRS platform and later processing them in software MS Excel, by using PivotTables.

In next few lines I would try to give introduction of mentioned company as well its main product – software AgenaTrader.

#### **3.1 Introduction and History of Include IT GmbH**

Include IT GmbH is software developing company founded in Vienna, Austria in 2010. Its main product is software for semi-automated stock exchange trading – AgenaTrader. CEO and founder of Include IT GmbH is Mr. Gilbert Kreuzthaler. Before starting this company in 2010, through his career he gained significant experience in two independent fields – stock exchange trading and C# programming language. This experience and combination of two professions were crucial human capital for founding Include IT GmbH. As long-time trader and programmer, he was able to see the gap in the market and start designing his own software solution – coding a software which would have all features that he would like to have as a trader.

Figure 16. Include IT GmbH, Hierarchical Organization



Besides Mr. Kreuzthaler, main development department is based in Ukraine with two C# software developers while rest of the team is based in EU, mainly in Austria and Germany. Vienna office have four additional departments: infrastructure, Quality Assurance (hereinafter: QA), Customer Support and Sales Department (Figure 16). Team consist also of numerous professional intraday traders which are valuable assets to company for their feedbacks and ideas regarding various AgenaTrader features.

Since observed subject is small enterprise, those departments relatively small - judging by the number of employees. For that reason some of them have very tight connection like, sales and customer support, customer support and Quality Assurance and Quality Assurance and development. This tight cooperation is useful because with common work some issues are solved faster. For example, any bug/problem that customer would report, if they are without solution, customer support is consulting firstly with Quality Assurance. If that attempt was not successful, Quality Assurance is consulting development in order to solve initial customer's problem. By doing this, company is trying to avoid usage of resourced dedicated for development and testing for two reasons: they are more expensive than sales support, and any involvement of those resources in solving user's issues is slowing down development. Simple communication is possible for team based in Vienna office, while other parts of company which are physically dislocated from headquarters have constant video link while everyday communication is done by Skype. Communication and cooperation from Vienna office departments and other, outside of Vienna have lower degree of efficiency than departments based in Vienna – due to lack of constant direct contact. Efforts are constantly done to erase any side effects of physical distance.

## 3.2 AgenaTrader

Center of attention are processes around software called AgenaTrader (Figure 17). This is highly sophisticated software designed for stocks/futures/forex trading. It offers to the traders manual, semi-automated or fully automated trading – this is actually the biggest advantage of this software. Since exchange markets (stocks, forex, futures and others) are quite dynamical, they force intraday traders<sup>4</sup> to spend each minute in front of their monitors in order to maintain their orders and positions. Besides that, they spent hours analyzing charts from numerous instruments on several time frames - all in order to find specific patterns which could be reliable signal for entry or exit point.

Using AgenaTrader, trader can turn that signal patterns in numerical algorithms, and load over 5.000 instruments at once and let software scanning the markets and notify him when specific market behavior is found. When specific pattern on market is found (signal), it can be transferred to trigger, that would start automatic trade – without any manual action of the trader. This software allows simultaneous scanning of multiple instruments on multiple time frames. Time frames are starting from a one tick (The minimum upward or downward movement in the price of an instrument), to day, week or a month – for any of those different trading strategy could be applied.

Figure 17. AgenaTrader Andromeda 1.8



Source: AgenaTrader Andromeda / Tradeescort.com, 2015b.

Besides scanning for signals, AgenaTrader have additional functions like managing stop orders. Stop orders in trading strategy are preventing losses if market goes in opposite direction than

<sup>4</sup> Intraday traders in most cases buy and sell the instrument during the same day by using timeframes shorter than one day

expected. When AgenaTrader's module AT++ is activated those stop orders are tracked and managed automatically – same concept goes for target orders. Without software help, traders are forced to move stop orders by themselves in accordance with market trends or they could leave them at initial level and risk - in that way they could miss profit opportunities. Automated managing of orders can be done in combination of 170 already built in market indicators. Users with advance programming skills can make their own conditions and indicators which later can be used as instructions for managing of stop order.

Size of orders and risk are key to successful trading. AgenaTrader also allows trader to set this parameters before trading, so each entry in market would be under previously set conditions. Up on each entry in market, software would calculate order size on basis of trader's account and situation on market. This helps to trader to have consistent strategy, which in long run should results in profits.

For any idea regarding trading strategy, instead of risking their own money, or spending time with paper-money accounts, back-testing module is developed. Each strategy that trader would like to use in future, before risking his own money, he can download historical data and apply his strategy on any instruments for any time frame and period. Doing this trader would save money (in comparison with testing with his own real money account) and time (in comparison with testing with demo account). In accordance with result, he can improve or completely dismiss his initial strategy.

Having mentioned most important features of this software, it is clearly that target group is not average computer user. Those are people with above average income, highly educated and with big passion for numbers, risk and stock exchange trading. Combination of this kind of user's characteristics makes quite high standards for quality. Any bug in software is potential financial loss for users – which is of course not refundable, market would not “forgive” traders for their badly placed orders and strategies. This is actually one of the reasons for this master thesis – demands for high quality code and means how to get and sustain that level of excellence.

### **3.3 AgenaTrader – AdminTool**

Since this is not freeware software, Include IT GmbH developed additional software for license managing. This is actually software which enable issuing and managing user licenses. Licenses are used for activation of various levels of functionality of AgenaTrader and its additional packages. AdminTool works on basis SQL database which consists of license keys and usernames.

Every time when contact with user is established, his “profile” is being updated with information like used operative system, brokerage / data feed, purchased packages. This information is useful in case of bug reporting. Knowing mentioned details is half job done



because testing department can significantly reduce number of testing scenarios for reproducing reported bug.

AdminTool is also very useful as statistical tool – it enables tracking of inflow of demo and purchased licenses and in that way it can present current trends to the management and using historical data, try to predict future trends as well.

## **4 BPM LIFECYCLE – INCLUDE IT GmbH**

Since observed entity is software developing company, its core process is software development of course. In accordance with that fact only core process would be audited. First task is to identify core business processes in Include IT GmbH and design them in form of “AS IS” BPMN model. To do that, all sources of information must be documented. This model should be workflow of all core processes that are running in this company. Afterwards, on basis of field experience and BPM theory model “TO BE” BPMN model would be derived. In execution part, process of implementation would be described. Implemented processes would be partially automated and partially we would upgrade existing one or completely eliminate them if they are unnecessary any more. Monitoring would be done for certain period of time, one week up to one month in order to see the new results of BPM intervention.

Quantities metrics would be applied where possible but preferred personal interviews will be conducted with CEO, as well with employees in order to find their opinion regarding changes in their business environment. In accordance with retrieved metrics and interview, final optimizing would be done.

### **4.1 Process Discovery and Analysis**

In order to define current state of business processes in Include IT GmbH, firstly is needed to define source of all inputs which drives development. As employee myself I was able to easily identify those sources during my regular working activities. By inputs it is considered all impulses that drives company’s processes forward – development of the software. Among them we have:

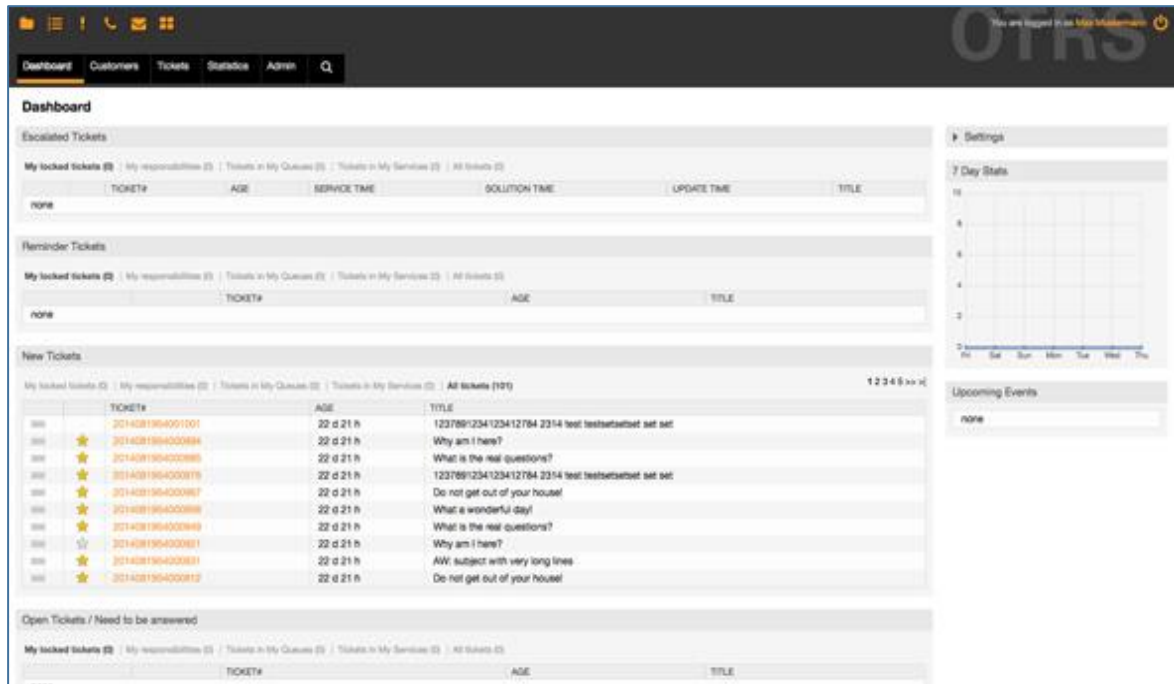
- OTRS.
- RedMine.
- Forum, emails and phone calls.

In second part, observed process would be the process of software release – their preparation, timing and impact of software quality.

### 4.1.1 OTRS

OTRS (Figure 18) is main infrastructure for dealing with customer's request. In most cases those requests are regarding support issues and sales questions.

Figure 18. OTRS



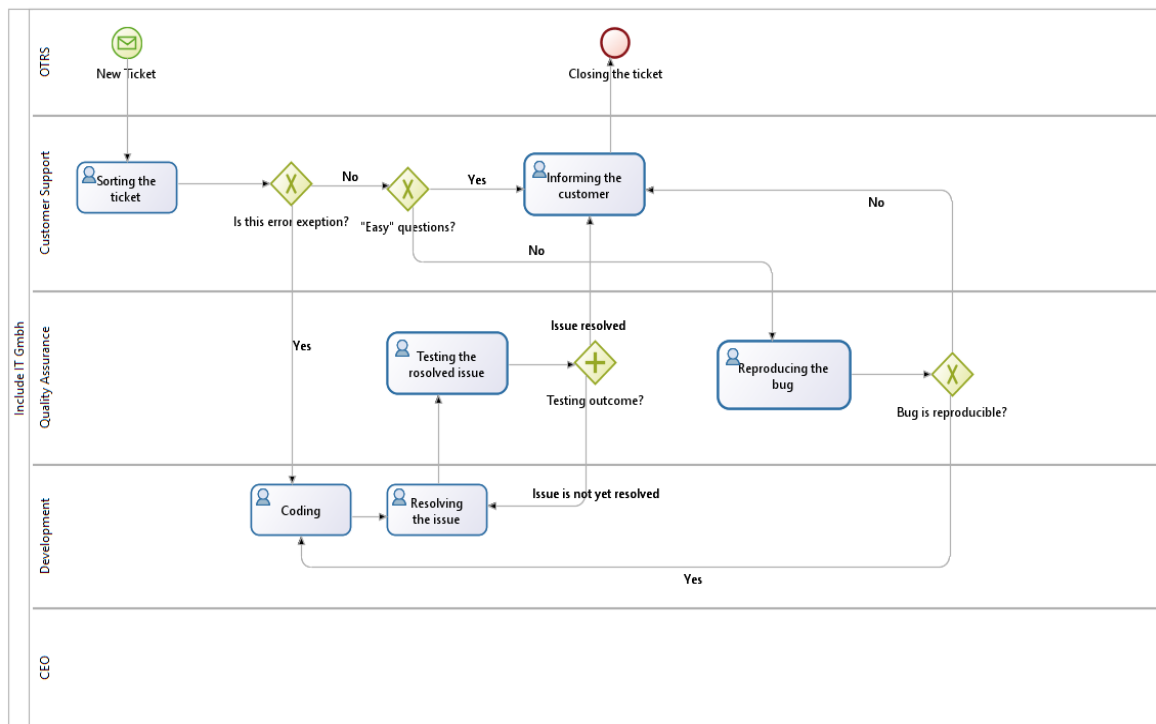
Source: Release Notes: OTRS 4 beta1 - [otrs.com](http://otrs.com), 2015.

OTRS is actually email platform which is managing emails in sophisticated way, enabling involvement of several employees at once while solving some user's issue. Employee responsible for OTRS is firstly trying to give proper support to customer and if he is not able to do that, then this request is forwarded to QA department where those issues are firstly tested in order to confirm their existence in current source code and then fixed by development team. If initial problem is confirmed in our source code then customer is notified with this results and he is given issue number so he can track progress of fixing process. After issue is fixed and tested with positive outcome, ticket is closed and customer is notified about outcome and version where new code would be implemented.

Same platform is often being used as help tool for executing internal decision from regular Monday meeting. In that way, stakeholders can track in any moment current state of confirmed decision. This decision start from small tasks as opening email account for a new worker, up to setting new website for a new web conference

As it can be seen from Figure 19, there are two main types of input information – error exception and user questions.

Figure 19. OTRS AS-IS Model “Dealing with Customer’s Request”



Error exception are automatically created in moment when software crash. Special module which is implemented in AgenaTrader tries to take screenshots of all open windows and makes memory dump which is submitted to OTRS platform<sup>5</sup>. This type of ticket is always automatically assigned to Developers – without any need for additional confirmation of issue. Depending from version of AgenaTrader priority is set – if that exception occurs on official release, than this is top priority issue because most likely all users have this potential bug on their machine, otherwise, if it happens on any of beta releases, it is set to second level priority as part of regular development activities.

Developers analyze memory dump in combination with screenshot and they implement additional code which should prevent reported exception. After Quality Assurance confirm the validity of implemented code, Customer Support is notified and end user as well

Main issue with current model is that is highly “manual work” intensive. This process could be only partly automated because very often it requires direct communication with end customers or potential customers.

#### 4.1.2 RedMine

<sup>5</sup> By sending automated email from user’s side, using Agena’s server and attaching all necessary information.

RedMine (Figure 20) is, just as OTRS an online platform, but it doesn't deal with emails. RedMine's main purpose is project management and issue/bug tracking. In background it uses MySQL, PostgreSQL or SQLite database.

Figure 20. RedMine

The screenshot shows the RedMine web interface. At the top, there's a navigation bar with links: Home, My page, Projects, Administration, Help. On the right, it says 'Logged as admin - My account Sign out'. Below the navigation bar is a search bar and a 'Jump to a project...' dropdown. The main content area is titled 'My project' and has tabs for Overview, Activity, Roadmap, Issues, News, Documents, Wiki, Files, Repository, and Settings. The 'Issues' tab is selected. On the left, there are filters for Status (open) and a dropdown for Add filter. The main table lists issues with columns: #, Tracker, Status, Priority, Subject, Assigned to, and Updated. A context menu is open over issue #79, showing options: Edit, Status, Priority, Assigned to, Copy, Move, and Delete. The right sidebar has links for 'New issue', 'Tracker', 'Issues', 'View all issues', 'Summary', 'Change log', and 'Custom queries'.

#	Tracker	Status	Priority	Subject	Assigned to	Updated
127	Bug	New	Normal	Ticket with attachments		12/22/2007 12:11 PM
116	Bug	New	Low	Keep playing audio when rw/ff and preserve pitch.	John Smith	12/17/2007 09:56 PM
88	Feature	Assigned	Low	HTTP Challenge-MD5 authentication		12/22/2007 04:33 PM
83	Feature	Assigned	Low	Export the parameters of an input	John Smith	12/17/2007 09:56 PM
82	Feature	New	Low	Formatted text rendering support	Dave Loper	12/17/2007 06:58 PM
81	Feature	New	Normal	DVTS support		12/17/2007 06:58 PM
79	Feature	New	Low	QuickTime capturing		12/17/2007 06:58 PM
78	Feature	New	Low	Full H323 videoconferencing		12/17/2007 06:58 PM
77	Feature	Assigned	Low	Closed captions / Teletext support		12/17/2007 06:58 PM
74	Feature	New	Low	Progressive download playing		12/17/2007 06:58 PM
73	Feature	New	Low	Dshow tuning support		12/17/2007 06:58 PM
72	Feature	New	Low	V4L tuning support		12/17/2007 06:58 PM
70	Feature	New	Low	Electric Program Guide		12/17/2007 06:58 PM
69	Bug	New	Low	SDL vout cleaning		12/17/2007 06:58 PM
65	Feature	New	Low	Protocol rollover support		12/17/2007 06:58 PM
64	Feature	New	Normal	Improve ZLM functionality		12/22/2007 04:33 PM
63	Feature	New	Low	Gstreamer and Helix integration		12/17/2007 06:58 PM
62	Feature	New	Low	Gnutella servlet		12/17/2007 06:58 PM
59	Feature	New	Low	Finalization of Pocket PC port		12/17/2007 06:58 PM
58	Bug	Assigned	Low	Re-write of the AppleScript bindings		12/22/2007 04:33 PM
57	Feature	New	Low	MacOS X SVCD support	Dave Loper	12/17/2007 06:58 PM
51	Bug	New	Low	Better Mozilla plugin control		12/17/2007 06:58 PM

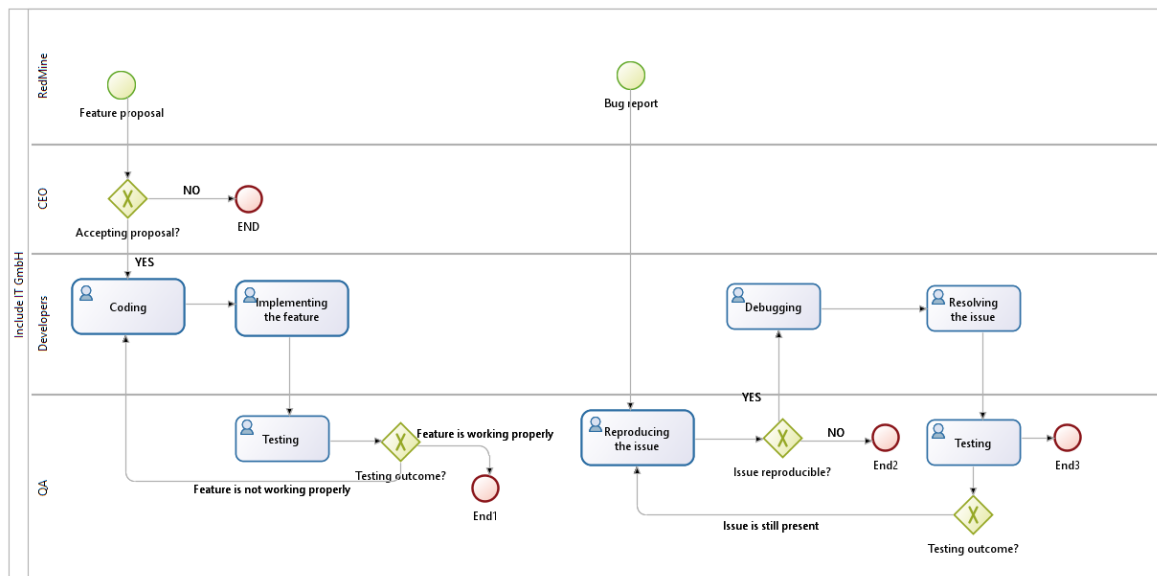
Source: Features - Redmine, 2015.

All bugs, feature request, proposals ends up in this system. Whenever CEO or any employee gets new idea about additional feature in software, which should be developed, he makes issue in RedMine and assigned it to CEO. In RedMine platform, term "issue" doesn't mean necessarily a negative connotation, it is a basic element for implementation which could be new feature or reported bug. If this issue could be valuable addition to final product, it is assigned to development department where this feature is later developed. Any issue enables attachments for additional explanation which is useful in cases of complicated explanation where we need video clip of steps how to reproduce the issue, or just presentation where is explained some new feature.

All issues can be assigned with next attributes: target version of implementation, urgency, OTRS number (if applicable), status and assignee. Attributes in RedMine platform are customizable so if needed, new can be implemented without any additional costs. Red Mine infrastructure in combination with attributes gives good ground for structured workflow. Since in RedMine we can track other user's (employees) activity, time needed for their reaction, number of assigned issues this could be good point for KPIs - starting from fixed bugs to successfully implemented features.

As it can be seen from Figure 21, there is certain difference from OTRS processes (Figure 19). While RedMine is used mostly for development or project management where CEO as decision maker is involved, OTRS is used just as operative level platform.

Figure 21. RedMine AS-IS Model “Project Management”



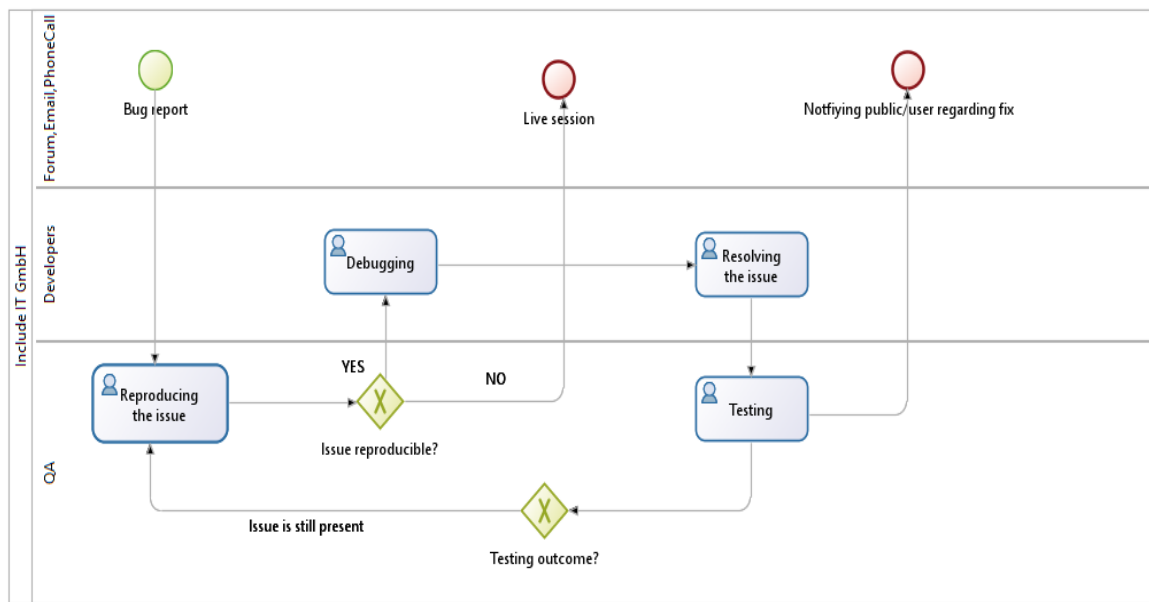
Two main operation are done over RedMine platform:

- **Implementation of a new features.** It is starting by making feature proposal and assigning it to CEO. Once in a month, CEO is listing all feature proposals and making plan for next version. Then, he is deciding up on assigned feature proposal. From that point he can reject it or accept it. Up on accepting it, he is assigning issue to developing team with proper target version and priority. After implementation is done, Quality Assurance department is getting assigned with initial issue to test the new feature. Depending from outcome of the test, whole workflow is ending or if feature is not properly implemented it is returned to development for additional coding.
- **Debugging.** After bug is reported, either from Quality Assurance or user's side, issue is assigned to Quality Assurance department in order to reproduce the issue. Reproducing issue is very important process, because only if proper pattern of actions is found, which results in bug, it can be fixed – otherwise this bug stays documented but not fixed. After founding proper order of actions which are causing reported bug, Quality Assurance department is making detailed description and instructions in issue and assigning it to development department. Depending from potential damage of this bug, specific priority is set. Development department then is debugging this issues and up on resolving it, it assign it back to Quality Assurance department. Then testing takes a place, and up on results issue is closed or returned to development.

### 4.1.3 Forums, Emails and Phone Calls

On company's website two separate forums are existing. Division is done on language basis, so currently exists English and German forum. Usually support staff is able to directly answer and help regarding user's question – in that way process starts and ends inside forum's borders and there is no need for further description in BPMN models. Since this software exist for several years very often experienced users give answers and advice to others – this detail is helping to reduce work load from customer support.

Figure 22. Forum, Emails and Phone Calls, AS IS Model



In cases when there is support issue which cannot be resolved by support staff, issue is being made in RedMine, and number of that issue is given on post so forum members could track development of issue. When bug is fixed and tested with positive outcome, issue in RedMine is closed and forum member in post is notified that he would have fix with next AgenaTrader update. Entire workflow is captured in Figure 22.

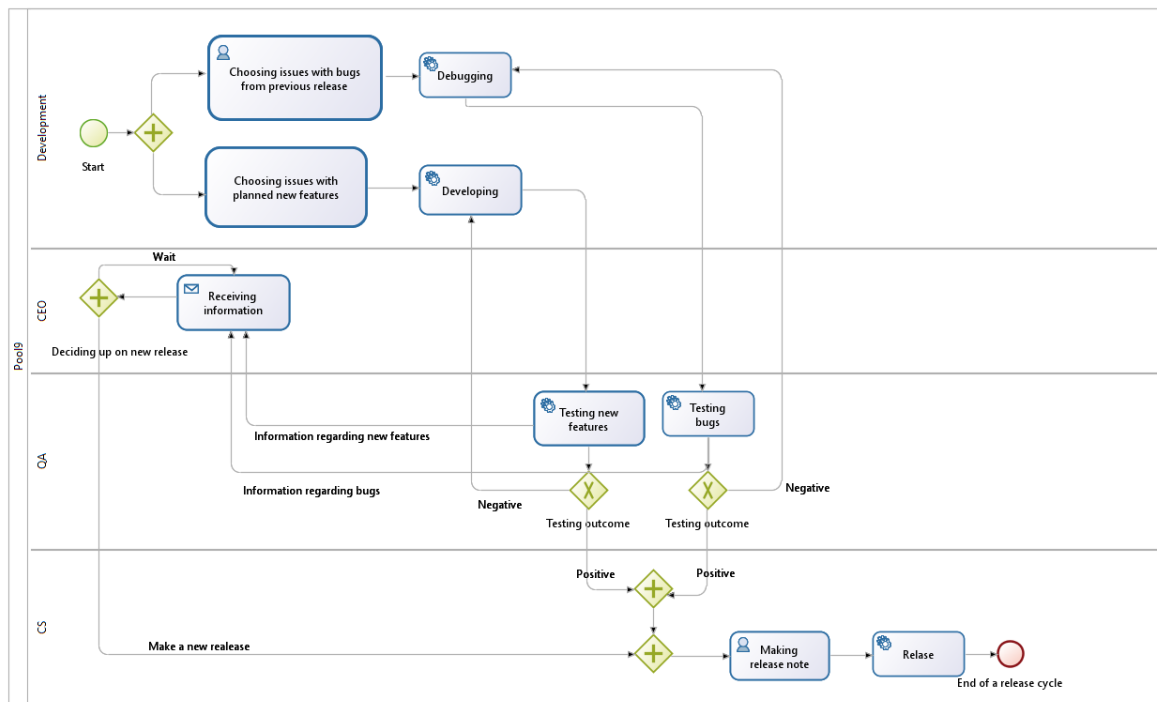
In similar way as forum, input of information works with emails and phone calls. Sometimes users choose email and phone calls over other methods because of privacy concerns. They would like to avoid revelation of their identity among other users in cases if they had some problems because this could affect their public image of professional trader. Maybe their reported issue is regarding showing some financial losses in software that they would like to keep for themselves or any other reason for which they would like to retain maximum privacy.

### 4.1.4 Software Release

Development works on bugs from previous release and new features and forward them to testing department. If bugs are found in previously processed issues, they are returned back to development, while other issues with positive outcome are closed.

Current process of software release is happening in time cycles in average of one to two months (Figure 23).

Figure 23. Software Release Cycle, AS IS Model



When desired level of fixed bugs and developed features is achieved, CEO announces that minor release would be made. This actually means that all users would get notification in their software to update to the latest release version. Before this decision is made, release note must be updated. Release note is a document which is following each version of software with detail explanations what is changed in that particular version. After that customer support department is taking Mercurial and in combination with RedMine (each entry in Mercurial is having reference number of issue in RedMine) is making new release note.

## 4.2 Process Redesign

In order to redesign existing processes several ideas were planned to be implemented:

### 4.2.1 RedMine subproject “BetaTesters”

In accordance with resources, company can employ only limited number of testing staff. Usually in testing departments significant amount of resources is saved by using automated testing. In

case of trading software which works with several other brokers where each one of them is using different API – automated testing is limited.

Option like outsourcing, would not give proper results, since testing of trading software can be done only by users which are skillful enough in both IT and trading. In same time there are several proactive AgenaTrader users, so project “BetaTesters” was next logical step.

Figure 24. Example of Release Note on BetaTester Project

[Edit](#) [Watch](#) [Copy](#) [Delete](#)

**AgenaTrader\_SetupV1.9.0.154.exe // short release note**

[< Previous](#) | 13 of 31 | [Next >](#)

Added by **Aleksandar Markelić** 14 days ago. Updated 14 days ago.

<b>Status:</b>	New	<b>Start date:</b>	02/15/2016
<b>Priority:</b>	1-Top Priority	<b>Due date:</b>	
<b>Assignee:</b>	-	<b>% Done:</b>	<div style="width: 0%; height: 10px; background-color: #ccc;"></div> 0%
<b>Category:</b>	-		
<b>Target version:</b>	1.9		

**Description** [Quote](#)

New AT is uploaded in DropBox where you can download it.  
 From previous release for this group (1.9.0.140) we did next changes:

Feature #6660: AgenaTrader Notifier is implemented. Currently only Android is supported and app is already in PlayStore. Who is interested to test it, please supply me with your email address that you use to access PlayStore. Solution for iOS is on the way but not in next two weeks.  
 Improvement done in Galaxy Chart Gallery #6288. Complete implementation is not yet fully done, but we expect to have it by 29. February  
 Improvements done in Watchlist #6290  
 Fixed bug regarding time synchronization pop up windows #6710  
 Fixed bugs #4657 #6776 #6742 #6769 #6574 #6752 #6652 #6519 #6786 #6765 #6247 #6246 #6080 #5478  
 Improved feature regarding multiselection in watchlist and AE #6764  
 Fixed bug regarding IBs order outside RTH #6573  
 Implemented auto mapping feature for Taipan. Right now it is possible to retrieve data for instruments when they are not mapped in Taipan, with condition that their filetype, symbol and exchange are properly set.  
 Implemented new feature regarding reverse AT position in DOM #5485  
 Implemented new broker and its mapping EXANTE #6732  
 Fixed bug with backtesting #5302  
 Implemented feature for futures, fractional view #6795  
 Improved MT-Pro phases  
 Improved volume chart

In next release, we would try finish refactoring, implement fully Chart Galleries (SingleChart Gallery, ListChart Gallery, TabChart Gallery)

In this version you should notice big changes regarding downloading speed of NTB charts like tick data, range and similar. This improvements is one of the results of refactoring which is in progress for some time. Datafeed/brokers where this improvements are done are Rithmic (rAPI+), CQG and GAIN / their APIs allowed this improvement.

We hope that we would get many feedbacks regarding possible new bugs regarding refactoring.  
 For any additional questions, I am always available on my skype aleksandar.tradesecort

**PS. When resolved issue is assigned to you, please give us feedback so we can close it or take additional actions.**

Main idea behind it is to make subproject of current RedMine where, besides current employees, beta testers would also have access. Beta testers would be selected by customer support department and send them invitation to join the project. They would not be paid for their participation but, they would receive top priority support regarding bugs they discovered, the newest features before official release and other incentives – modest version of “Bug bounty program” like Microsoft, Google and Facebook have. In return for incentives, AgenaTrader would try to gain from their trading experience, additional security and stability improvement as result of additional bug checking from their side. Since they are experienced traders, new ideas for future development are expected as mutual gain. Main plan is, to release the latest beta each second week. In that way all bugs that Quality Assurance department missed to catch, beta tester would/should find them. So, if first beta release is given on 1th in the month, by the 15<sup>th</sup> in the month new bug reports by Quality Assurance and beta testers would be fixed and parallel new features would be developed. Each time when beta release is given, proper release note is given for beta testers, so they are aware what changes are done in last two weeks (Figure 24).



## 4.2.2 New Support Contact Form

Current support is mostly based on direct email correspondence. In this process Include IT GmbH is losing significant amount of working hours. Firstly to retrieve all needed information regarding reported issue, used broker and data feed and other needed information in accordance with reported problem. Following this workflow, time spent from initial report up to solution can last over one month.

*Figure 25. Support Contact Form*

PRODUCTS SHOP DOWNLOAD PARTNERS SUPPORT E-LEARNING ABOUT US TESTIMONIALS f g t

### Client Support form

All fields with (\*) are mandatory fields

Your Name*	<input type="text"/>	Operating System* (Win XP & Vista are not supported)	---
Your E-Mail *	<input type="text"/>	Broker*	---
Subject*	<input type="text"/>	Datafeed*	---
Category*	---	Account*	Broker - DEMO
AgenaTrader Release*	---		
Is it a one-time occurred or repeatable issue? *			Repeatable
At which time frames does an issue occur?*			<input type="text"/>
With which instrument does an issue occur?*			<input type="text"/>
Please describe an issue more precisely. What should have happened and/or what had occurred instead?*			
<input type="text"/>			
Please send us your screenshots, so that we could gain a better understanding and avoid possible misconceptions.			
<input type="button" value="Browse..."/> No file selected.			

Source: *Contact Support / Tradeescort.com, 2016b.*

There are two main reasons for this amount of spent time – first is previously described long process of obtaining all necessary data and second one is that we are getting more and more users and with them more customer support requests.

With increasing tempo of inflow of new users, where 90% of them are demo and roughly 10% are paid users, new working force is required. Even though licensed users always have priority, demo users must be served too, because otherwise, they would never decide to purchase the software. If would be decided to hire more workforce in customer support it would not be profitable, because, tempo of demo user request is growing more than licensed one, which means that newly hired employee would not contribute to incomes which they are being spent.

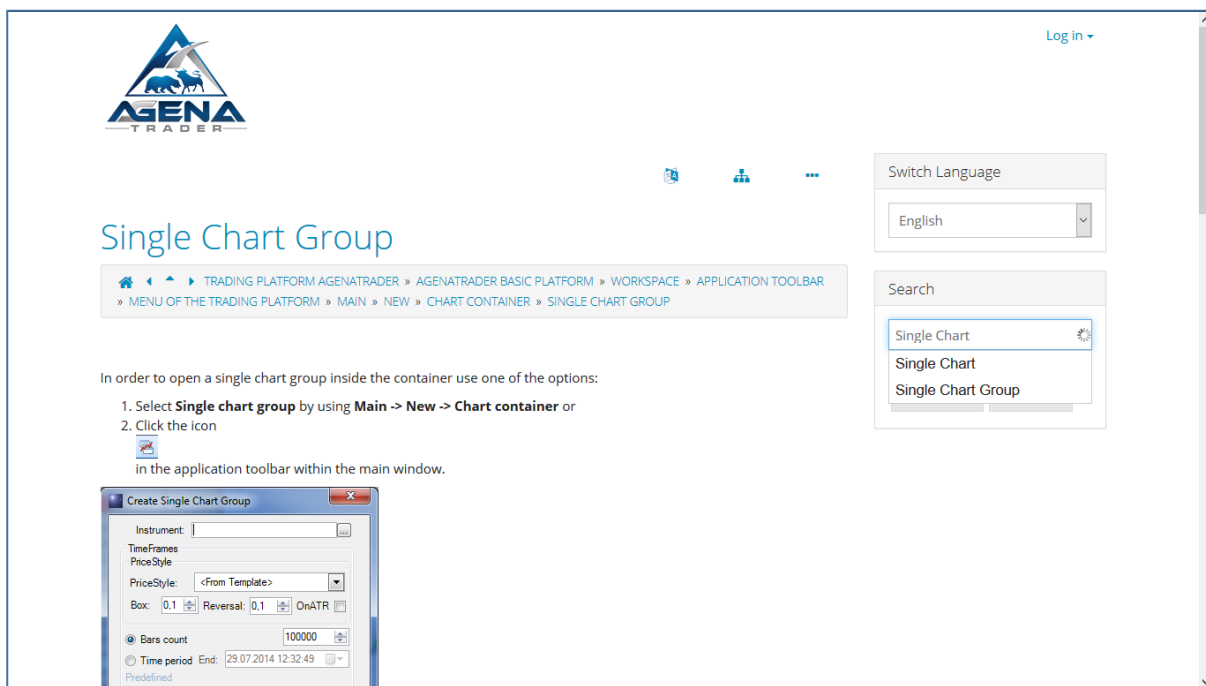
One of the ways to increase efficiency of customer support department is new support contact form (Figure 25).

Result of this implementation should be that from first instance we are getting all necessary data and we can focus on solution in first response. Since every broker or datafeed is specific, those data is required from start, as well version of AgenaTrader, description of an issue, and screenshot or any other attachments. In this way responding and resolving time would be decreased, which should result in user satisfaction and trust in AgenaTrader.

#### 4.2.3 AgenaTrader Wiki Project

In order to provide clients with the best quality help files, Wiki project of AgenaTrader is implemented (Figure 26).

Figure 26. AgenaTrader Wiki Project



Source: AgenaTrader Wiki / HomePage, 2016a.

In well know fashion as Wikipedia works, all help material is sorted in AgenaTrader Wiki project. This help users to navigate with ease and quickly find answer for their questions. Result of this implementation should be decreased pressure on customer support department. Expectation are that when quality online help content is provided to public, that before sending any requests to customer support, help files would be read and answer would be found there.

#### 4.2.4 AgenaTrader Education

Project which would create whole new ecosystem around AgenaTrader. Since more than a half user requests are caused by poor knowledge of software, it is decided that independent project

would be implemented - AgenaTrader Education certification. Education would be based on structured tutorial videos which would be divided in modules – in accordance with every AgenaTrader feature. During the videos and at end of each module, user would have to answer certain set of question in order to pass the module. This concept is well known today and it would not be some kind of prototype, because on Internet we have many online education website like for example Coursera. All modules from one level must be passed in order to acquire that certificate. This system would have three levels of education – basic, advanced and expert. Basic would be free of charge and it would prepare user for correct usage of all features. Education project can be used as leverage in future, when users are coming with some general questions regarding product, they could be forwarded to education project. Then customer support would have one more additional element for prioritization – request from user with the highest level of certificate would have higher priority than one with lower or any education certificate. This method would be based that people without any knowledge about software are most likely asking how to activate some feature, which should definitely belong to basic education. Advanced level of education would deal with specifics of technical features regarding defining mathematical formulas for detecting particular market signals. This level would require live intervention of mentor and therefore they would not be free of charge. Expert certificate are earned when all previous are already passed plus if user/trader is making significant success on market and have his own trading style. When user acquire this level, he would be listed on AgenaTrader Education certificate website, as one of certified mentors for our new users. This would give him opportunity to hold education seminars/webinars with AgenaTrader approval – and of course have additional income from this channel.

#### **4.2.5 Key Performance Indicator implementation**

Two main KPIs are planned to be implemented in order to avoid future bottlenecks. First one should be on Quality Assurance/development side, which should enable better precision in estimating date of completion of one product (implementation of features + correction of bugs). This could be done by measuring number of open and closed RedMine issues by each AgenaTrader version. Next step would be to check issues assignments', who's duty they are and for how long they are not processed. To do so, RedMine issues can be exported daily in regular CSV file which could be later processed by using simple MS Excel PivotTable functions. Using RedMine information, in combination with Pivot Charts and trendiness could give us better picture about future expectation. Tempo of decreasing open issues, and increasing closing issues could show some indication regarding activity in development and Quality Assurance department. Of course, whole measurements which are based on RedMine issue number have one serious flaw – two issues do not mean necessary that they require same amount of work in development/testing. We have no information regarding working hours which should be far more important information when making estimations regarding future release.

Second KPI would be exclusively on customer support side, measuring performance of that department. That could be done by measuring average time needed for closure of OTRS tickets. If two indicators would be used, one as 3 last month's average time and last week average time, than we could also see future trends of performance and prepare on time if we need more personnel in any of departments.

#### 4.2.6 Build With Tags

Previously, when executable setup file was made, there was no method to find the date at which exact commit build was started. Commit represent any new code implementation – new feature of bug fix. Only estimation could be done on basis of executable date of creation. So looking at Figure 27 for example, if there is a version that is four days old, that would mean ten possible outcomes - since there is ten commits on that day when executable setup file was made. Outcome leaves place for speculation regarding final commit of observed executable and thus additional work load for Quality Assurance.

Figure 27. Mercurial

Graph	Rev	Branch	Description	Author	Age	Tags	Phase
	10897+	V1.8 Hotfix	Not a head revision! ★ Working Directory ★	Aleksandar	now		
	10908	default	default tip Added tag 1.9.0.77 for changeset 312f550be124	Build Server	12 hours	tip	public
	10907	V1.8 Hotfix	Bug #5605 "Dukascopy Error"	Anatoliy Dudarchuk	19 hours		public
	10906	default	1.9.0.77 Bug #5605 "Dukascopy Error"	Anatoliy Dudarchuk	19 hours	1.9.0.77	public
	10905	V1.8 Hotfix	Feature #6294 "IB: Level 2 data from DOM"	Anatoliy Dudarchuk	20 hours		public
	10904	default	Feature #6294 "IB: Level 2 data from DOM"	Anatoliy Dudarchuk	20 hours		public
	10903	feature/history manag...	feature/history manager refactoring Feature #6355 "History Manager Refactoring (NTB, OFA, etc.)"	Anatoliy Dudarchuk	27 hours		public
	10902	default	Added tag 1.9.0.76 for changeset 5c7c97136674	Build Server	36 hours		public
	10901	default	1.9.0.76 Added tag 1.9.0.75 for changeset 045f854f19cb	Build Server	2 days	1.9.0.76	public
	10900	default	1.9.0.75 Added tag 1.9.0.74 for changeset 4e6ec09599b	Build Server	3 days	1.9.0.75	public
	10899	default	1.9.0.74 Added tag 1.9.0.73 for changeset d0f9407c8187	Build Server	4 days	1.9.0.74	public
	10898	V1.8 Hotfix	Added tag 1.8.0.424 for changeset b80fd95776bd	Build Server	4 days		public
	10897	V1.8 Hotfix	1.8.0.424 Merge	Nazar	4 days	1.8.0.424	public
	10896	V1.8 Hotfix	Gain Synchronization.	Nazar	4 days		public
	10895	V1.8 Hotfix	Feature #6510: Price Histogram - Step 4	Nazar	4 days		public
	10894	V1.8 Hotfix	Feature #6510: Price Histogram - Step 3	Nazar	4 days		public
	10893	V1.8 Hotfix	Feature #6510: Price Histogram - Step 2	Nazar	4 days		public
	10892	V1.8 Hotfix	Feature #6510: Price Histogram - Step 1	Nazar	4 days		public
	10891	default	1.9.0.73 Bug #6517 "Dukas Copy / Trades"	Aleksandar	4 days	1.9.0.73	public
	10890	V1.8 Hotfix	Bug #6517 "Dukas Copy / Trades"	Aleksandar	4 days		public
	10889	V1.8 Hotfix	TaiPan Issues: NullReferenceException(ForceUpdateLastQuote), NullReferenceException(Subscribe)	Nazar	5 days		public
	10888	default	TaiPan Issues: NullReferenceException(ForceUpdateLastQuote), ArgumentOutOfRangeException(Bu...	Nazar	5 days		public
	10887	V1.8 Hotfix	Support #6511: TaiPan problems with London stocks (difference of GBP to GBX)	Nazar	5 days		public
	10886	default	Added tag 1.9.0.72 for changeset d48b3a118128	Build Server	5 days		public
	10885	default	1.9.0.72 Feature #6510: Price Histogram - Gain Forex zero histogram issue	Nazar	6 days	1.9.0.72	public
	10884	V1.8 Hotfix	Feature #6513: Demo request button in license window	Nazar	6 days		public

## filter text ##

AgenaTrader.Helper/Handlers/DatafeedDomHandler.cs  
 AgenaTrader.IBDatafeed/Consts.cs  
 AgenaTrader.IBDatafeed/IBDatafeed.cs

Changeset: 10905 (1b7eb4084cfc8) Feature #6294: IB: Level 2 data from DOM  
 Branch: V1.8 Hotfix  
 User: Anatoliy Dudarchuk <dudarchuk@gmail.com>  
 Date: 2015-12-07 19:44:00 +0200 (20 hours)  
 Parent: 10898 13f92af34405f1 Added tag 1.8.0.424 for changeset b80fd95776bd

This makes several new possibilities to test when making testing scenarios. Second problem was that, executable setup files were built only up on explicit request towards CEO to make a new build. Often it happens that CEO is busy or he forgets so all tests regarding requested build are waiting. With this simple issue, additional delays are created for testing scenarios and many working hours are wasted in Quality Assurance.

Connection between version number and commit is important for Quality Assurance department and for development because, each bug that is reported, in order to be debugged, must be reproduced in development environment. For that purpose, new tagging mechanism is

implemented – with each build, automatic or manual, special tag is being assigned to commit for complete tracking (Figure 27 – vertical red rectangle).

With implementation of this simple tool, general overview of complete project is improved for all departments in company.

### 4.2.7 Description of Reported Bugs

Previously bugs were reported without notation about version where they occurred. With this bad practice in power it was almost impossible to track down this issue after few days, weeks, months. The problem was that, when software is developing, each day we are building new version and after some time we are unsure is this issue still present in the latest release.

Figure 28. Example of Additional Description in Redmine Issue

IB / TWS 954.20 (09.12.2015)

[« Previous](#) | 24 of 399 | [Next »](#)

Added by Aleksandar Markelić 3 months ago. Updated about 2 months ago.

Status:Closed

Priority:1-Top Priority

Assignee:Aleksandar Markelić

Category:Datafeed/Broker

Target version:V1.8

OTRS ID:

Start date:12/10/2015

Due date:

% Done:0%

Spent time:-

Description

Quote

Tested with 1.9.0.78

We are having problems with newest TWS platform.

-In cases where TWS 954.20 is installed, when we start it over AT, we are starting old TWS  
-where only TWS 954.20 is installed, we are failing to start TWS at all  
-when we start manually TWS 954.20 and then run connection to IB we are getting next error msg:  
Exception: System.NullReferenceException. Message: Object reference not set to an instance of an object.. Stack Trace: at AgenaTrader.IBDatafeed.IBDatafeed.cRU=(Object chU=, updateAccountValueEventArgs cxU=)

TAKA027\_IBFA.log (42.2 KB) Aleksandar Markelić, 12/11/2015 09:10 AM

12-11-2015 09-10-23.jpg (843 KB) Aleksandar Markelić, 12/11/2015 09:10 AM

Subtasks

Add

Related issues

Add

History

Updated by Aleksandar Markelić 3 months ago

#1

- Description updated (diff)

Updated by Tolik Dudarchuk 3 months ago

#2

- Status changed from New to Resolved
- Assignee changed from Tolik Dudarchuk to Aleksandar Markelić

Updated by Aleksandar Markelić 3 months ago

#3

- File TAKA027\_IBFA.log added
- File 12-11-2015 09-10-23.jpg added
- Status changed from Resolved to New
- Assignee changed from Aleksandar Markelić to Tolik Dudarchuk

Tested with 1.9.0.80

Still not working (12-11-2015 09-10-23.jpg; TAKA027\_IBFA.log)

When that kind situation happens, issue is returned to Quality Assurance department where it is being tested with the latest version to be sure, do we still have this issue. This is of course tremendous waste of time. For that reason, “rule of good practice” was implemented regarding process of reporting the issue. If issue is being reported with executable that reported issue should always start with: “*Tested with [VERSION NUMBER]*” (Figure 28) or in cases where testing was being done with developing environment: “*Tested with dev env 10905 (1b7eb4084cf8) Feature #6294 "IB: Level 2 data from DOM"*” (example from Figure 27–horizontal red rectangle). In this was additional information is given, it is clear was the issue

reproduced in executable or development environment where original source code was compiled in MS Visual Studio and ran after compilation is done

With new practice, when developer starts debugging the issue, he can immediately see was the problematic part of the code already affected by some other fixes and in that way, save significant amount of time for QA department.

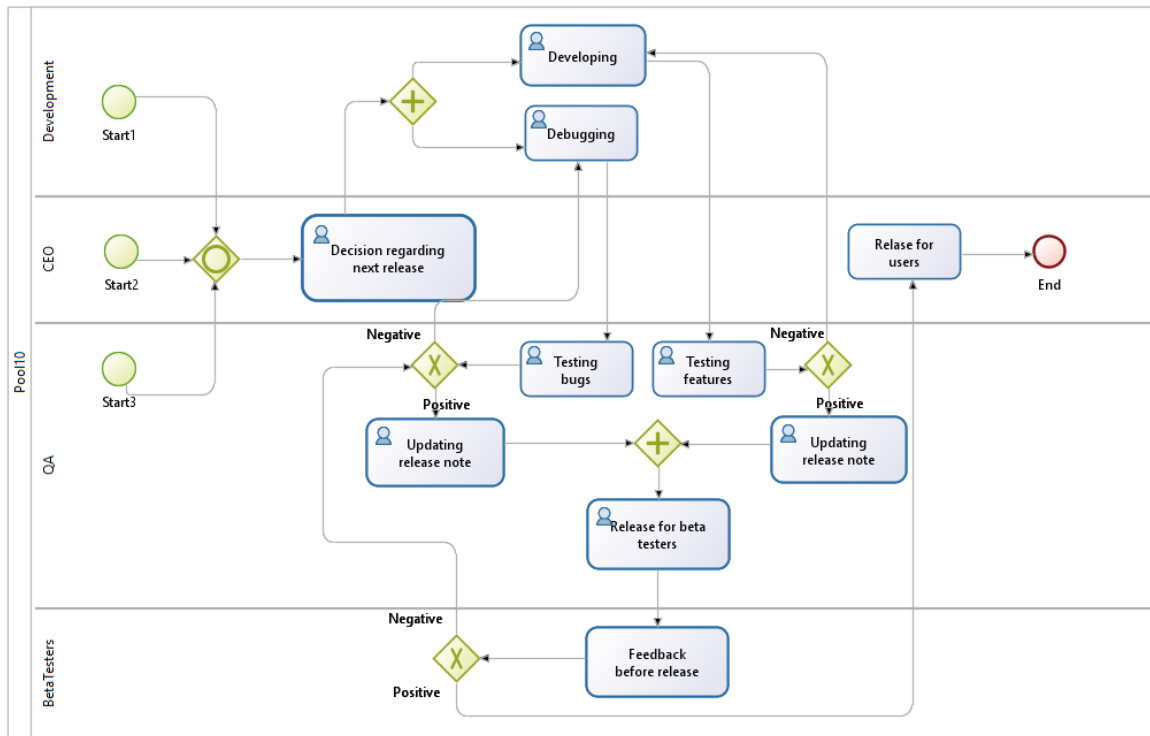
#### **4.2.8 Software Release Cycle**

Three main changes are planned in software release cycle and those are:

1. Changes regarding selection of the issues which would be processed for next release.
2. Process of building of release note.
3. Additional ring of software stability – including beta testers.

Decision regarding issues which would be included in next release is not anymore brought by just development department. Before this change, development was selecting most convenient or any of high priority issue and worked on that. Even productivity was not the issue, problem was that management did not had any (or minimum) impact on planned development – it was impossible to predict what would be fixed/developed by next week or two. Now CEO, Quality Assurance department and development are making decision on basis of plans for future releases – in best case scenario this means plan for next two weeks. This change is mainly caused by need in future to develop two or more product simultaneously with same team and then project management would be seriously challenged with previous structure. Release notes are not done any more by customer support department, they are done by Quality Assurance department in moment when issue is tested with positive outcome. Example of that process can be seen in (Figure 29), where Quality Assurance department is making regular updates regarding official version. Regular users do not get RedMine issue to read, but instead of that, customer support department is using this base to deliver final form of release note – rich HTML stylized environment.

Figure 29. Software Release Cycle, TO BE Model

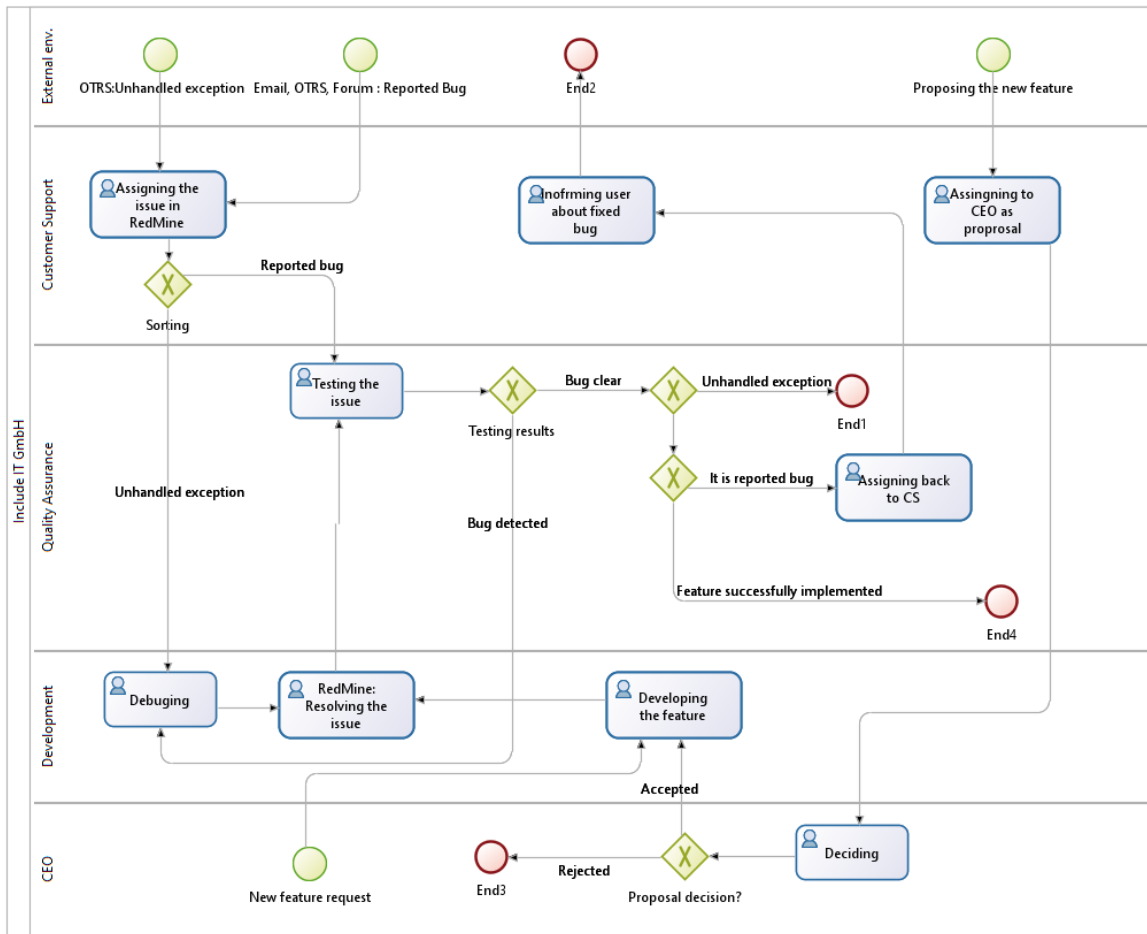


Reason for that change is fact that Quality Assurance know more about closed issue in that moment, than customer support after one month – time needed for completing this process is severely decreased because it is done during whole cycle while before, it was done when release decision is made. Final change is included beta testers as additional security ring. Before going out with release, it is given to beta tester to use it at least for one week. Because, each fix which is done in official version (1.8 in this case) is almost always implicitly included in 1.9 beta as well. In that way, as soon as we go out with beta release (with current tempo, each two weeks), almost all bug fixes for 1.8 are tested in that cycle as well. In accordance with collected feedbacks we are making final decision for releasing new version of software.

#### 4.2.9 Bug Fixing/Feature Developing.

Stated ideas would significantly decrease load on customer support department and, what is more important, we are expecting increased end buyer satisfaction. Decreased load on customer support means as well more time in Quality Assurance to work on stability of software instead dealing with issues caused by lack of knowledge to correctly operate AgenaTrader. Compiling all this concepts in one holistic solution, we get TO BE BPMN model shown in Figure 30.

Figure 30. TO BE Model of Bug Fixing and New Feature Developing



In current TO BE model we see three different type of inputs from external sources. We see unhandled (error) exception which are being sent automatically if AgenaTrader crash. For all automated bug reports we do not inform client regarding their fixes. Bugs of course, would be fixed and the new code would be available up on next update, but no special feedback information is given like when we get manual report.

Second source would be bugs reported from our forum, emails and phone calls (and later also by BetaTesters). This issues first comes to customer support where they have first check. In many situation there was just problem with some settings or wrong interpretation from user's side.

For reported problems for which customer support establish that could be genuine bugs, they are forwarded to Quality Assurance. There reported bug is closely examined and efforts are invested to reproduce<sup>6</sup> them. In best case scenario bug is reproduced, and Quality Assurance is giving detailed instructions to development regarding scenario which is causing bug or in special occasions, if developer is available, debugging is done right away over live session in Quality Assurance employee's computer.

<sup>6</sup> Reproduce, replicate, repeat reported bug in development environment like MS Visual Studio 2013



Third and fourth source of issues are actually feature requests. The one which comes from users and employees must arrive to CEO first for final decision, afterwards, if positive, they are being moved to development. Customer's proposal are also important but there is problem with "noise". Very often those proposals are some user wishes that just are not possible or require too much developing hours, but sometimes they truly represent some valid need of users. It is up to customer support to try to filter valid requests and forward them in RedMine. Of course, CEO's request are going directly to developing. After everything is set in order, we can move to next phase – process implementation.

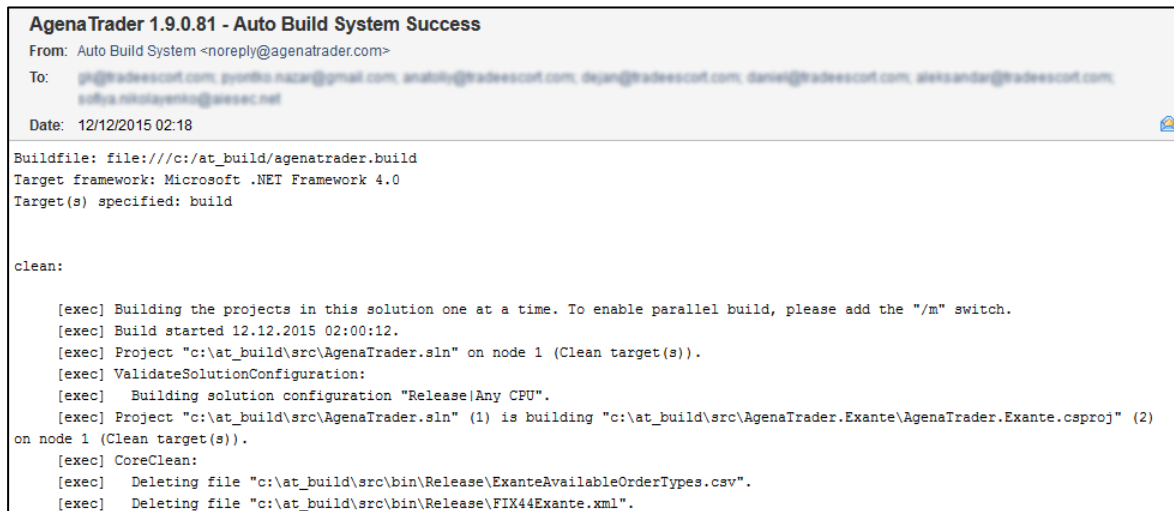
### **4.3 Process Implementation**

Depending up on complexity of project, implementation of particular processes are planned into agenda. First step was to implement platform for all other phases of execution.

In order to closely track development of all implementation and resolution of all internal and external tasks "3M" was introduced – Monday morning meeting. As the same names says, each seven days, all team members have meeting with CEO. It starts with informal breakfast where CEO and employees are just discussing in relaxed atmosphere. Later, as breakfast comes to end, CEO firstly state points that he would like to address, those can be internal or external nature. For each one of them, responsible persons are dedicated and deadline for which task should be resolved. Some of those points are just information regarding new contracts that company made. In second part employees are bringing they own points that they would like to address and get some resolution. This small upgrade is contributing a lot for creating healthy company's culture and team building. Since CEO have consultative management style, all decisions brought at this meeting have none or very small resistance. It must be emphasized that this method is still feasible only because of relatively small number of employees. As number of employees is growing this meeting would be hard to maintain for two main reasons: increased loss of working hours and decreasing ability to achieve consensus. But at this point of company's development it is suitable for its needs – it is excellent platform for executing all changes in company, building healthy company culture and future monitoring of any project.

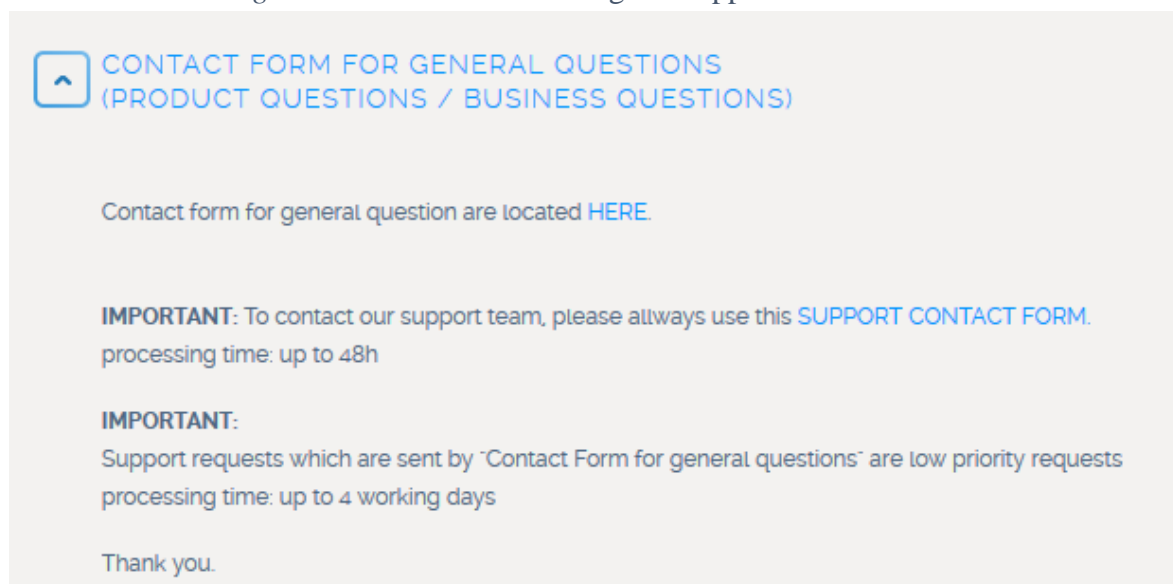
From this point we plan to lunch execution of all ideas in accordance with time needed for their complete implementation. Changes regarding core process of reporting the bugs and "builds with tags" are not demanding from any perspective and for that reason they are implemented first. Employees in Quality Assurance got complete instruction how to continue in future workflow while for builds, developers made a script. Developers made small extension of original idea - for beta version (in this case version 1.9.x) build is created automatically each day at two o'clock by just using simple \*.bat file and windows schedule module and notification email is sent to selected employees (Figure 31). In that way testing of the newest implementation in 1.9 can go faster than compiling the code from scratch. Originally planned tags are implemented for both versions 1.8.x and 1.9.x.

Figure 31. Auto Build Email



Next step is forming the Beta Testers group. Enabling technical platform is easy part, because actual platform already exists (RedMine), we need only to create additional subproject. More challenging part was selecting people who would actually be part of Beta Testers project. It is decided with CEO that all partner traders would be included in this project and particular users which are selected by customer support. Those users have been most active on official AgenaTrader's forum where they showed significant level of enthusiasm and proactivity. People responsible for this project are partially customer support and partially Quality Assurance. Even this represent additional load for those departments, we are expecting trade off in means of increased quality of code, early detection of bugs as well ideas for new features.

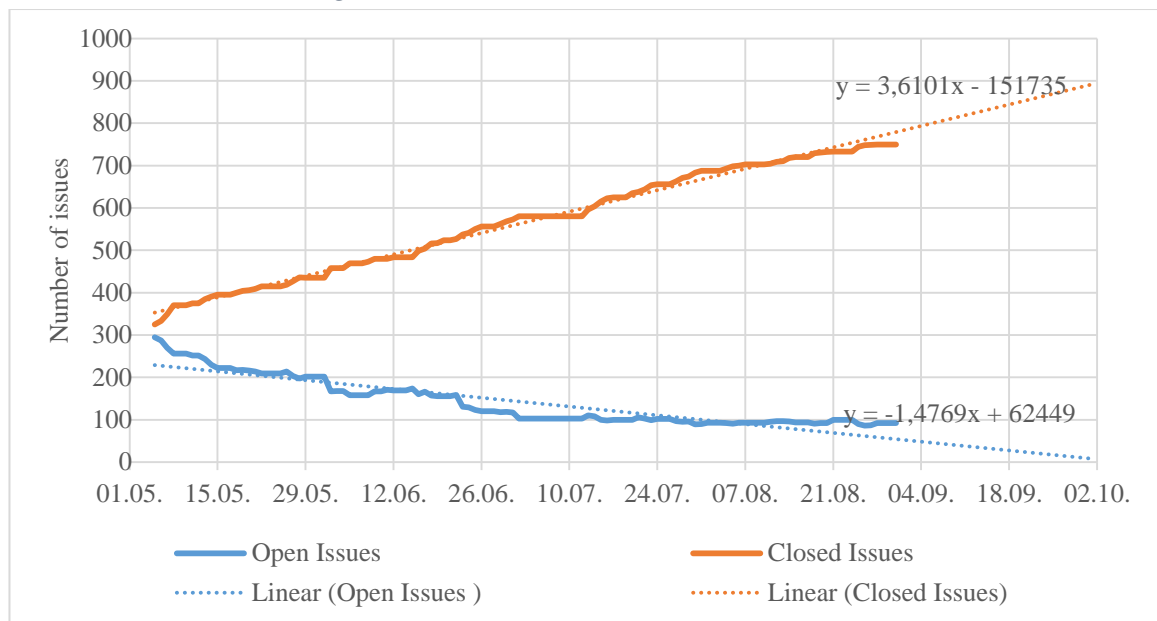
Figure 32. Additional Warning for Support Contact Form



Source: *Contact / Tradeescort.com*, 2015a.

Customer support would get additional relief in their workflow, because of mentioned new support contact form (Figure 25). This web form is “feeding” OTRS database with additional data. Each entry is creating new “client profile” on basis of its email address or updating existing one. In that way for customer support it’s easier to have clear picture even without additional contact with customer. For customers which are concerned regarding their privacy or they refuse to enter any data for any reason, old contact form is still enabled. In order to stimulate users to use new form, additional notice is given (Figure 32). In this way, if user is trying to place request using old form, his is warned that his request would be set with low priority and that it could take up to 4 working days to process.

Figure 33. RedMine KPI / Burndown Chart



KPIs implementation is done in combination of OTRS and RedMine. RedMine’s open issues are firstly listed per project and then exported in Comma Separated Values (hereinafter: CSV) file format. From that point processing is done over MS Excel and its Pivot function. Most important categories are number of open and closed issues per project. From that point additional data can be collected regarding last day of activity on an issue. In that way some “forgotten” issues can be spotted and their solution can be reinitiated or in the worst case scenario they can just be closed without solution. In some particular cases, AdminTool data can be used in combination with other key indicators like number of new demo licenses for last month in correlation with dynamics of official web page visits. Mostly used RedMine KPI is measuring completion of one product (Figure 33).

Number of issues are recorded for each day – open and closed one. Chart that is developed for this purpose is actually customization of burndown chart. Difference from actual burndown chart is that, there fixed date is set, when one project would be finished, while here, we are making estimations. As we can see from chart, on basis of previous tempo since 1th May to 1th September, we are estimating that we would complete project by first week of October.

Modification of that kind is done due to size of company and its agility. Same employees who works on development, sometimes they are interrupted because unplanned change in API of some of the brokers, or they need to concentrate on some critical bug fixing etc. If development department have higher number of employees (currently we have only two + CEO) we could dedicate half of them to work for development and other half for emergency and debugging. When this phase of growth comes, we could transform to classical Agile Scrum methodology easier than then now.

Wiki Project execution would not take long time, basically old link for help would be just replaced with new one, but due to its database size, its building process it would take some time. Linking would be done in executable of AgenaTrader as well on its website. Eventually when all data is collected, additional time would be invested so this data base could be available in German as well in English.

At the moment of writing of this thesis, AgenaTrader Education project is still in building phase. Significant amount of working hours must be invested in building tutorial videos, together with quiz questions which needs to available to English and German language.

#### 4.4 Process Monitoring and Controlling

Monday morning meeting is main platform for monitoring all of implemented processes. As data source, we would use RedMine and OTRS. For RedMine two main KPIs are developed: average days of delay and sum of days of delay. Methodology is next: open issues are firstly exported in excel where they are processed in next manner: last update date is subtracted with current date (Table 4).

Table 4. RedMine KPI / Bottlenecks (14 December 2015)

<b>Assignee</b>	<b>Nr. of Issues</b>	<b>Avg delay</b>
Employee #4	106	96
Employee #7	21	12
Employee #9	21	24
Employee #3	16	94
Employee #2	13	40
Employee #1	9	28
Employee #5	6	82
Employee #6	3	32
Employee #8	1	11
<b>Total</b>	<b>197</b>	<b>70</b>

Values are calculated for each issue and later used for calculation of sum and average per employee Final result is shown in form of Pivot Table. For privacy reasons names are generic in

form of “Employee #”. As it can be seen from Table 4, employee #4 have 106 issues with average delay of 96 days per issue. At first sight situation is alarming, but we must be aware of special circumstances. Currently major refactoring of complete source code is in progress - for that reason employee #4 (which is developer and CEO in same time) cannot work on his issues.

Other employees can be analyzed with different criteria. Interesting result can be, for example, employee #3 - his is having 16 issues with average 94 days of delay. In this situation CEO or any other responsible person, with simple double click can inspect all assigned issues for this employee and make conclusion is this really alarming situation and how to solve this bottleneck.

Table 4 could be good KPI indicator for future, where we could inspect average days of delay per each week (each Monday) and by that monitoring overall performance of whole company. Same methodology can be done for OTRS, where mainly customer support would be monitored while RedMine is used for all departments.

In accordance with monitoring results of bottlenecks, we could either modify existing processes or, if we are left out of any solutions, we could try to employ additional worker. If situation from Table 4 continues to exist after refactoring is done, CEO should reconsider for some changes regarding number of employees or redistributing existing work. Of course, this should be done if it is concluded that employees with high average days of delay is not slacking – otherwise company could get in situation to unwillingly promote unproductive behavior.

Second major KPI of software quality is number of unhandled exception in official release. Each report of that kind is actually failure of Quality Assurance department and BetaTesters project.

## 5 PROPOSALS FOR FURTHER IMPROVEMENTS

While writing this thesis, several elements appeared which could be improved beside already implemented improvements.

Five levels of priority can be set for each issue in RedMine from its early starts, but this functionality is currently abused. Since there is no clear guidelines and definition what can have the highest priority, very often employees used this attribute to push some issues purely on basis of their feeling towards some bug. Results of described behavior is that roughly 26% of open issues have the highest priority (Immediate) and contradictory they have higher delay for 3 days than average of all issues (Table 5). In practice this situation would mean that if some issue is critical, that setting its priority to highest would not help at all, but maybe even postpone its fixing. Setting all issues to the highest priority issue would not result that they are all important, but quite the opposite – that none of them is.

Table 5. Bottlenecks with All Priorities, (15 December 2015)

Assignee	Nr. of Issues	Avg delay
----------	---------------	-----------

Employee #4	106	94
Employee #9	21	26
Employee #2	16	34
Employee #7	15	14
Employee #1	14	32
Employee #3	6	104
Employee #8	3	5
Employee #6	3	32
Employee #5	3	17
<b>Total</b>	<b>187</b>	<b>67</b>

Looking at low priority issues (Table 6), situation is quite the opposite – average delay is smaller for 22 days than the highest priority.

Table 6. Bottlenecks with Only Low Priorities, (15 December 2015)

<b>Assignee</b>	<b>Nr. of Issues</b>	<b>Avg delay</b>
Employee #4	8	65
Employee #9	5	42
Employee #7	4	16
Employee #1	1	99
Employee #5	1	19
<b>Total</b>	<b>19</b>	<b>48</b>

In comparison with highest priority issues (Table 7), low issues have share of only 10%.

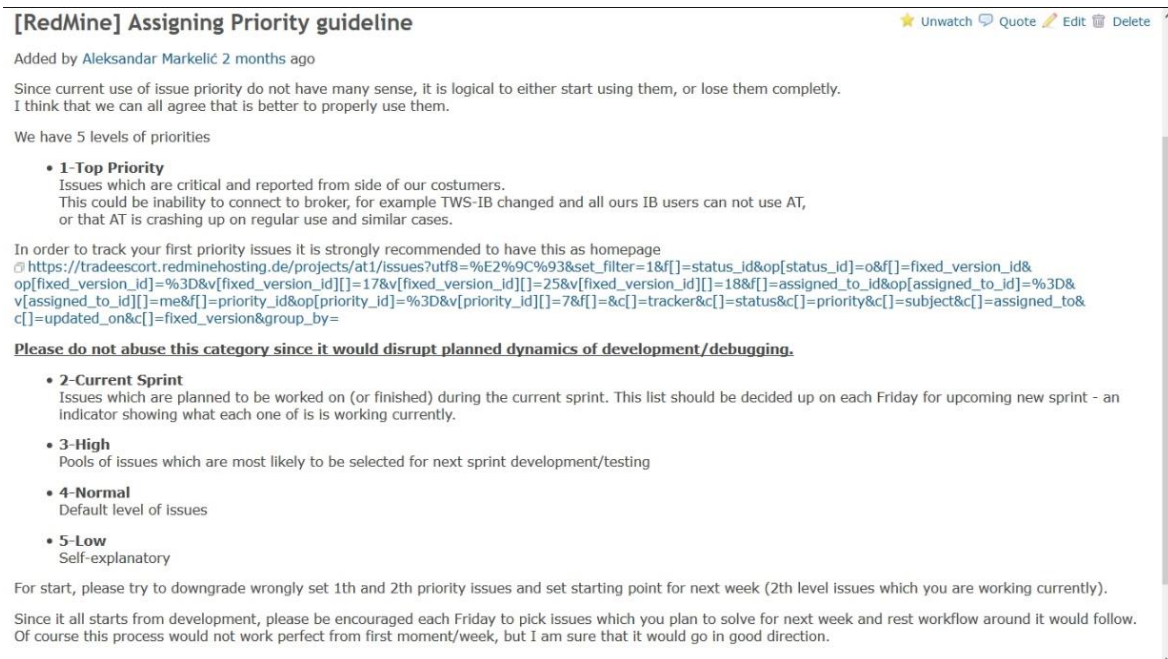
Table 7. Bottlenecks with Only Immediate Priorities, (15 December 2015)

<b>Assignee</b>	<b>Nr. of Issues</b>	<b>Avg delay</b>
Employee #4	28	103
Employee #2	8	19
Employee #9	4	9
Employee #1	3	8
Employee #3	2	124
Employee #8	2	7
Employee #6	2	47
Employee #5	1	29
<b>Total</b>	<b>50</b>	<b>70</b>

Situation of that kind, is slowly leading company in “slow response” mode and could affect poor performance while solving customer reported critical bugs. Company still do not feel the side effects of this bad practice because it has small number of employees which are regularly in contact, each one with another – so they explain directly how important assigned issue is. In that way by direct contact they explain to development what actually highest issue is and make an argument for it, but if company intends to grow, this segment of corporate culture would need to

change. Psychological effect of this simple metrics should not be dismissed. Best example for that are employees #5 and #3 – on 14<sup>th</sup> December they were provided with their stats (Table 4) and information that they would be tracked in future. In just one day employee #3 finished 10 of his issues (62,5% of all issues assigned to him), while employee #5 completed 3 issues (50%) and decreased average delay from 82 to 17 days.

Figure 34. Assigning Priority Guideline



Wrong conclusion should not be made, this doesn't not mean that constant productivity would jump for some enormous percentage like 50-60%, but only that employees would be more careful regarding their duties, especially if their stats are noticeable (in negative manners). For stated facts, active monitoring should definitely be established in future where employees (in)activity would be tracked and evaluated.

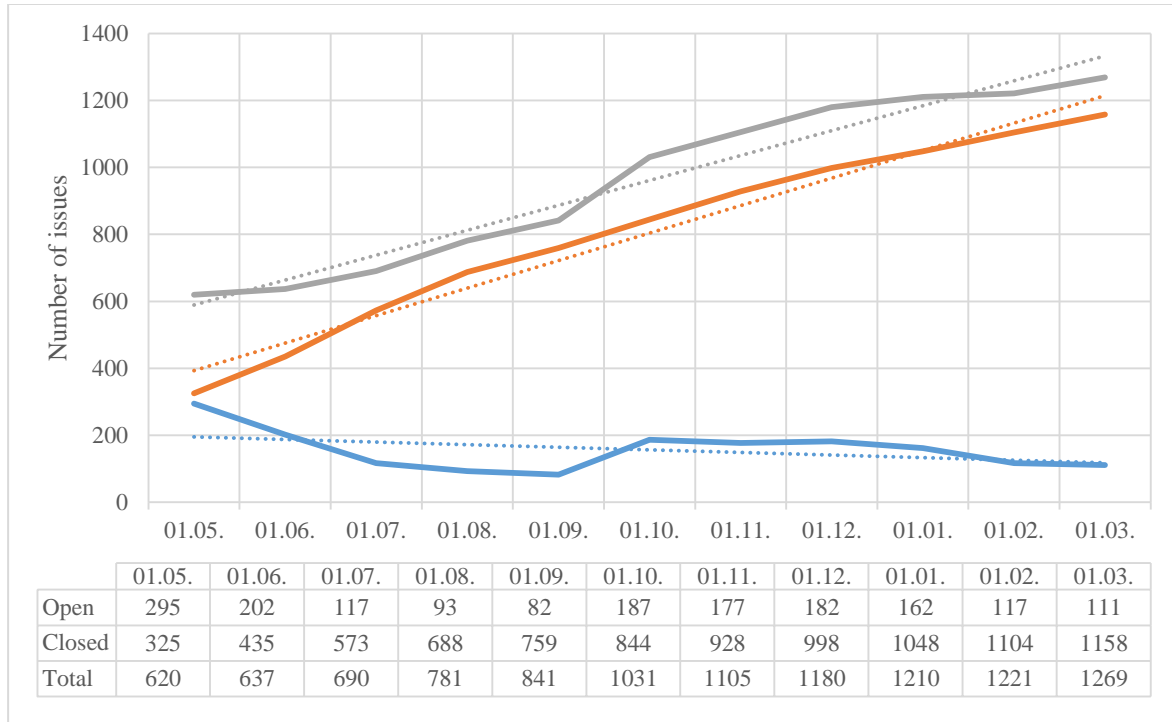
Reason for this behavior is that, all employees do not have clear goals to achieve week by week (self-motivation is main guide for work), and then, they choose to work on less “boring” issues neglecting in that way for several days issues which are assigned to them. Without proper metric, only way to spot this behavior would be actual delay for new official release or feedback from customer side – asking for estimated time needed for fixing bug that they reported. Second case is extremely harmful for the company public image because it could ruin trust in whole product. So far this behavior did not caused any serious side effects, but it could do so in near future. In order to prevent or at least minimize this kind of behavior, small guide line is written for future use (Figure 34):

Next suggestion would be regarding KPI - burning down chart. During development of this master thesis, named KPI was made as well. From Figure 33, which was taken on 1th September, on basis of current tempo of closing issues and number of opened one, 1th October



was estimated date when all the issues from 1.8 version would be fixed. The problem is that today is 20<sup>th</sup> March (approximate half year after Figure 33 was made) and there exists still over 100 issues in 1.8 (Figure 35).

Figure 35. Redmine 1.8 Issues per Months



Reason for this strange situation lies in fact that many issues appeared after 1st September. By using this chart we can see that total number of issues (open and closed combined) skyrocketed from 841 at 1st September 2015 to 1.210 in February of 2016 - in average 76 new issues each month. This is normal behavior for any software when release is made ( Myers & Putnam, 2003, p. 55). But even this metric is not complete explanation since starting from May to September, there is also upwards trend regarding total number of issues - 55 new issues per each month. One of the logical explanations is that in time period, more resources are shifted to develop 1.9 version – so even with almost same tempo of increasing new issues, number of open issues have risen because they are not being processed at that time. This is best example why ad hoc project tools cannot work properly in cases where multiple projects are running simultaneously. Second explanation is that, more users started using 1.8 so they covered more testing scenarios than Quality Assurance could. By doing so, new bugs are discovered and documented in RedMine.

In this case, as it can be seen in Figure 35 issues are fixed in accordance with their tempo. Only critical period that can be spotted are September and October when number of open issues broke current trend, but afterwards, dynamics of issues was under the control. This is also due to BetaTesters project, because in this period project started and part of those issues is from their side as well. Reported issues during that period were not critical so no serious damage was done to the public image of the company, but to be able to monitor several projects, IT Include GmbH



needs to think about general solution which would combine Agile/Scrum with IT project management – new platform which would replace current RedMine, or at least upgrade its current workflow.

Currently just two projects are making already trouble with estimations, even though very often they have same improvements (all improvements from lower version are simultaneously improvements of higher version as well), but additional problem would represent dimension of mobile solutions. In near future it is planned to develop mobile solution for three operative systems – Android, iOS and Windows. Each one of those projects would have issues dedicated only to their ecosystem. Beside difficulties in estimations of completions, whole new problem would be how to synchronize some releases of several projects.

## CONCLUSION

Initially, when I started writing this thesis I knew that I would face some unorthodox solutions, since BPM is not so common in IT companies mainly for agile nature of whole industry. Due to my professional inexperience I was expecting some radical changes like Michael Hammer described in Ford case ( 1990, p. 107). But, in Include IT GmbH relatively small changes could be done, looking from general overview of processes.

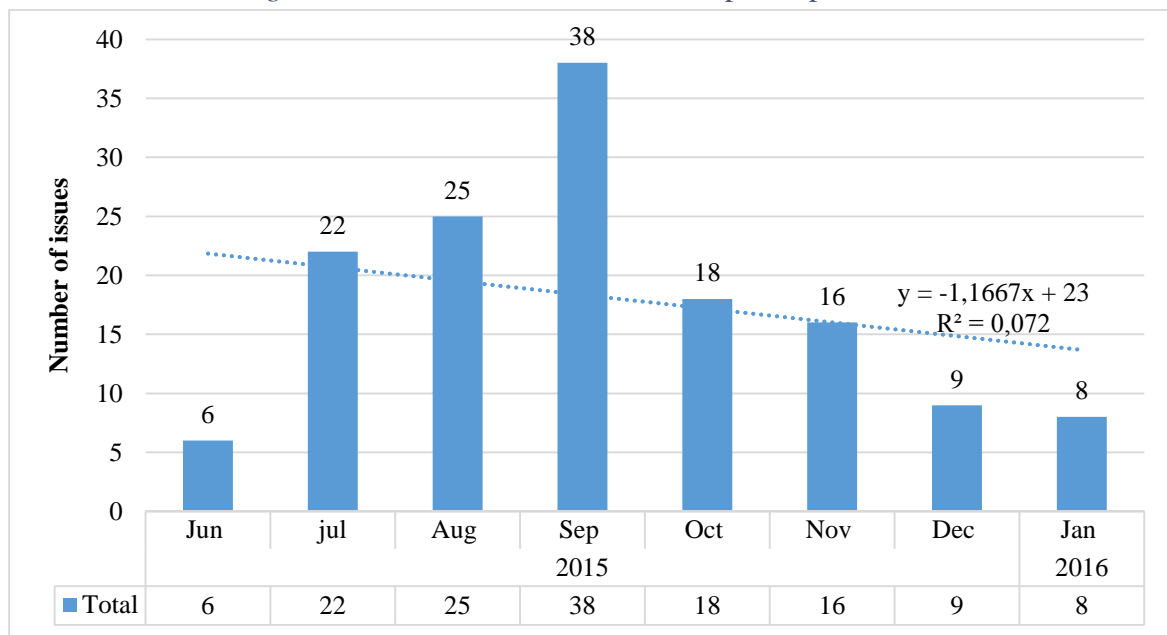
Full potential of Bonitasoft couldn't be applied since management was not keen to experiment with core changes of their business processes. Main reason for that decision lie in fact that efforts invested into implementation of Bonitasoft would be soon wasted because company's final goal is solution which is completely Agile/Scrum compatible, which Bonitasoft is not (or it requires significant number of working hours to be developed in that stage). Only usage of Bonitasoft could be done in role of BPMN model builder. While executing those tasks, drawing AS IS and TO BE model, it was noticeable that Bonitasoft is made for more than just BPMN. Connection of processes, ability to easily save/load models and representative look are clearly not top priority of Bonitasoft. BPMN in this case was used just for programming – and not for representation like it was done in this master thesis. So, it could be said that Bonitasoft in this case was misused.

Significant impact on final result are done by modifying some “micro” processes like describing new issue, making auto-build process, designing form for contacting support – in all cases their goals of those processes are achieved, workload on employee's side is decreased, leaving thus more time to deal with another tasks. Even it was not main purpose of it, motivation was risen by implementing Monday Morning Meeting and KPIs. With establishment of regular meeting in order to track and implement changes in company, positive side effect is achieved as well team building. Employees really got feeling that their opinion matters and they saw that is possible to make changes in workplace if they make suggestion and other agrees up on it. KPIs from another side, is increasing motivation with different way – regular tracking of each employee

performance is definitely positive impulse to keep issue list clean as possible. Company is still have place for improvement which was obvious from Table 5.

Regarding quality of the code which is delivered to end users, BPM did the most. Implementation which are responsible for increment of quality are BetaTesters and enhanced software release process. Proof for this statement can be in fact that AgenaTrader 1.8 was officially released in June of 2015. As users had transited from previous version 1.7 to official 1.8, they started reporting issues which QA failed to notice (Figure 36).

Figure 36. Number of OTRS Cases Opened per Month

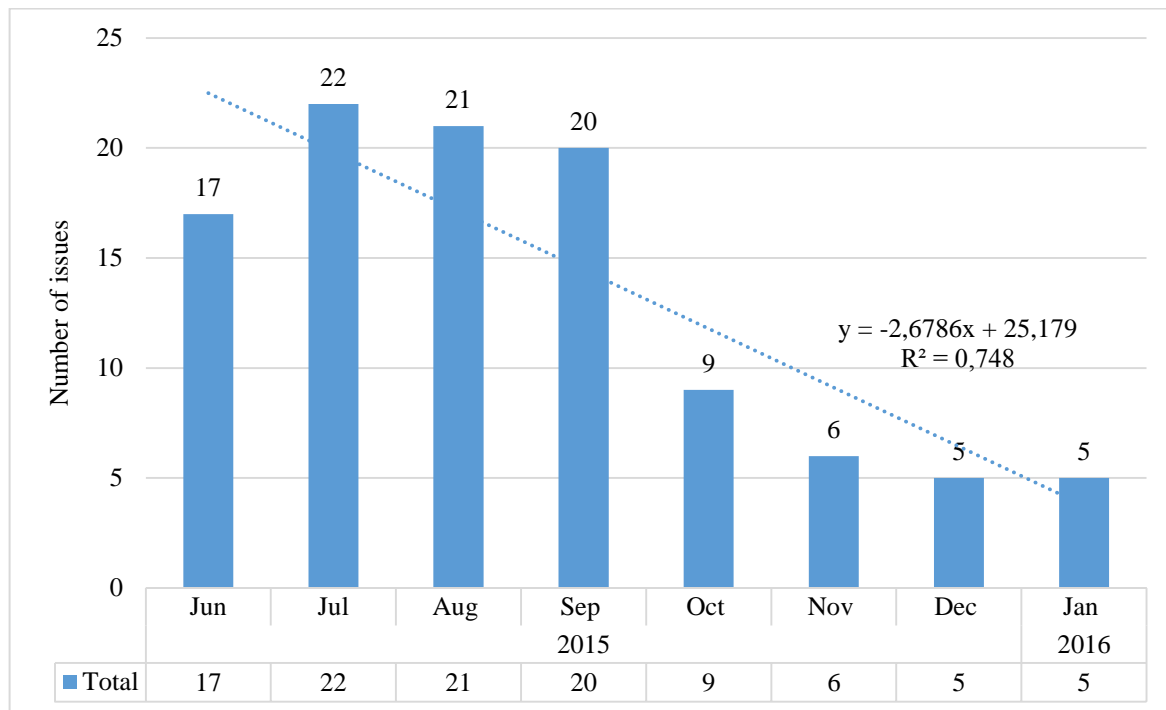


This can be seen in the chart, that each month, more and more new issues from final users were reported – from just six in June to thirty-eight in September. Starting from that month, first effects of project BetaTester is noticeable. As reported issues are fixed, they are still checked with old methodology plus additionally given to Beta Testers to try before general public. As previously mentioned, this covers significantly more using scenarios than Include IT GmbH can afford to pay for testing. Experience from IT industry would suggest of course to use automated testing scenarios, but in case of AgenaTrader that option is not possible – to be more precise, it is possible but it would cover only small and insignificant part of software while its core (connection with datafeeds/brokers, correct execution of trades and strategies) would still require manual and exploratory testing.

Similar behavior can be seen with unhandled exceptions (Figure 37) which are sent automatically up on each crash of AT. We can see that starting from official release, AgenaTrader 1.8 was not stable, and this state lasted for next four months. Only when new improvements regarding quality started to work, significant drop of unhandled exceptions is noticed – in average, twenty new exceptions per each month before quality implementations vs six new exceptions per each month after implementation. However this doesn't not mean that

starting from October AgenaTrader is having 50% less unhandled exceptions – explanation is slightly different. Since all 1.8 features and fixes are in same time 1.9 changes, then any bug which is made it is detected firstly in 1.9 by QA and BetaTesters. By doing that, official release stability is increased on contrast of 1.9. Conclusion is clear, starting from October, just like in Figure 36 significant drop of reported issues is obvious – quality implementations gave positive outcome.

Figure 37. Number of Unhandled Exception Cases Opened per Month



Looking from CMM perspective, Include IT GmbH made improvement. At beginning of this paper, this company was standing in repeatable level<sup>7</sup>. Main techniques of project management are already established in form of RedMine infrastructure, certain processes already existed and their documentation could be seen in RedMine's history. Element which is almost putting this company back to initial level of CMM is its CEO which is in same moment main developer for AT++ module of AgenaTrader. CEO role itself requires many tasks to be done, internally and externally and combining CEO role with main developer role is risky but yet only adequate strategy in current situation. Since programing in C#, in financial industry and for creating trading strategies is very rare skills to have in one person<sup>8</sup> and there is no one in current team to replace him– this compromise was forced to be done, at least for the moment. During the writing of this paper Include IT GmbH achieved one level up – defined level. Core processes got detailed structure with defined stakeholders, starting from inflow of bugs and new features up to publishing new release online. Documentation regarding all changes are captured in RedMine

<sup>7</sup> Second out five, where fifth is the best one

<sup>8</sup> All named skills are necessary to work on AT++ module

and as comments in source code<sup>9</sup>. Regarding problem with main developer/CEO, solution was found in form of “support window”. Support window meant that one day in a week (in this case, Thursday) is devoted just for debugging/developing while during the other days in week, old practice is in force. In that way, dynamics became more stable and more appropriate for planning, because for each week there was minimum developing hours from main developer that company could count on.

For better overview, of complete situation, interview with CEO is conducted, where he was being asked to comment changes in company in last half year – time for which paper was in process.

Gilbert Kreuzthaler:

“Since we started with tracking issues from OTRS and Forum, users requests are managed way much easier back and forwards. Possibility to establish clear tracking from start to end give us clear methodology to responds to users up on resolution of their bug reports or feature proposals.

Regarding BetaTesters, they are one of our best improvements regarding code quality. At very early stage of development we are getting high quality feedbacks – all because users from this project are using our beta version as productive one. Those early adopters are one of our big advantages, since with them appearance of critical bugs in final release is minimized. Users from BetaTester group are using beta release way earlier before final release and this gives us quite good feeling regarding future expectations. Besides Forum and social networks this is our tightest connection with our users where we truly function as community – improving significantly customer satisfaction.

Changes which are done in Mercurial, like tags and auto building are not helpful with quality, but it is more time-saving upgrade. It help developers and me as well, to easily locate bugs and determine if they are maybe already fixed in current release. Developer’s working hour are precious because they are engine of this company, for that reason this is small but efficient upgrade.

Issue tracking is useful but not fully functional solution and in this stage it represent “half-truth”. It could be considered as first step toward ideal stage where company would be able to track several parallel projects, applications, plan their releases in accordance with available resources. Final goal of company should be SCRUM implementation in IT project management which would be customized for our needs.

---

<sup>9</sup> All coding lines which are starting with “//” are comments. They are used as explanation and reminder why some programing method is used.

Project of Agena Education and Wiki is very important since it encourages users to self-educate in using AgenaTrader and of course in trading. It should “force” people to learn and effectively solve their problems without contacting customer support. Project is still in progress and we cannot see the results, but we are expecting gradual increase of customer support time savings.”

On basis of CEO’s interview it could be said that BPM is truly never ending process which runs in cycles. Even at this moment new improvement in processes are being done – in direction of SCRUM and improved IT Project management.

Business process management in Include IT GmbH without doubt improved customer satisfaction (over better quality and stability of software due to BetaTesters, faster issue resolution due to prioritization, community effect due to BetaTester project), decreased workload of development and QA department (due to advanced tagging in Mercurial) and customer support (due to Wiki, enhanced web form). Final conclusion on basis of CEO’s interview and metrics done in this thesis is that changes which are done, gave positive results. In contrast to all read papers, all changes (except Bonita) were easy to implement because corporate culture of employees (and CEOs) was completely oriented towards innovations – this gave excellent environment for business process improvements. BPM successfully improved IT Include GmbH but that is not reason to stop there, but only to continue with optimization and future improvement with increased strength from acquired new experience.

## REFERENCE LIST

1. Aalst, W. M., & Song, M. (2004). Mining Social Networks: Uncovering Interaction Patterns in Business Processes. In *Business Process Management* (pp. 244-260). s.l.: Springer.
2. Aalst, W. v., Hofstede, A. H., & Weske, M. (2003). *Business Process Management: A Survey*. Eindhoven: Springer.
3. Agile Alliance. (2001). *Principles behind the Agile Manifesto*. Retrieved April 30, 2016, from <http://www.agilemanifesto.org/>
4. Agile Alliance. (2015). *What is Agile Software Development?* Retrieved August 23, 2015, from <http://www.agilealliance.org/the-alliance/what-is-agile/>
5. Bhatt, K., Tarey, V., & Patel, P. (2012, May). Analysis Of Source Lines Of Code(SLOC) Metric. *International Journal of Emerging Technology and Advanced Engineering*, 2(5), 150-154.
6. Bonitasoft. (2015). *Bonita BPM Engine architecture*. Retrieved August 9, 2015, from <http://documentation.bonitasoft.com/bonita-bpm-engine-architecture-3>
7. Bonitasoft. (2016). *See an application in the Bonita BPM Portal*. Retrieved March 20, 2016, from <https://www.youtube.com/watch?v=FO9l7Zv9Qc8>
8. Capterra. (2016). *Best Business Process Management Software | 2016 Reviews of the Most Popular Systems*. Retrieved April 30, 2016, from <http://www.capterra.com/business-process-management-software/>
9. China Internet Network Information (CNNIC). (2016, February 4). *The Internet in China: How the Network is Shaping the PRC's Economic and Social Landscape*. Retrieved March 1, 2016, from <http://www.worldfinancialreview.com/?p=4978>
10. CMMI Institute. (2010). *CMMI for Development Quick Reference*. s.l.: CMMI Institute.
11. CMMI Product Team. (2006). CMMI® for Development, Version 1.2. *Software Engineering Institute*, 1-55.
12. Cockburn, A., & Williams, L. (2000). The Costs and Benefits of Pair Programming. *Extreme programming examined*, 223-247.
13. Dumas, M., La Rosa, M., Mendling, J., & Reijers, H. A. (2013). *Fundamentals of Business Process Management*. Heidelberg: Springer.
14. Gartner Inc. (2014). *Gartner Says 4.9 Billion Connected "Things" Will Be in Use in 2015*. Retrieved March 3, 2016, from <http://www.gartner.com/newsroom/id/2905717>
15. Gartner, Inc. (2015). *Business Process Management - BPM - Gartner IT Glossary*. Retrieved August 16, 2015, from <http://www.gartner.com/it-glossary/business-process-management-bpm/>
16. Habib, M. N. (2013). Understanding Critical Success and Failure Factors of Business Process Reengineering. *International Review of Management and Business Research (IRMBR)*, 2(1), 1-10.
17. Hammer, M. (1990). Reengineering Work: Don't Automate, Obliterate. *Harvard Business Review*, 68(4), 104-112.

18. Harmon, P. (2005). Service Oriented Architectures and BPM. *Business Process Trend*, 3(4), 1-4.
19. Harmon, P. (2010). The Scope and Evolution of Business Process Management. In J. vom Brocke & M. Rosemann (Eds.), *Handbook on Business Process Management 1* (pp. 37-81). Heidelberg: Springer.
20. Humphrey, W. S. (1989). Managing the Software Process. *Testing the Data Warehouse Practicum*, 1-15.
21. IBM. (2016). *What's new in IBM Business Process Manager V8*. Retrieved August 16, 2015, from [http://www.ibm.com/developerworks/bpm/bpmjournal/1206\\_pacholski/1206\\_pacholski.html](http://www.ibm.com/developerworks/bpm/bpmjournal/1206_pacholski/1206_pacholski.html)
22. Include IT GmbH. (2015a). *Contact | Tradeescort.com*. Retrieved August 9, 2015, from <http://www.tradeescort.com/en/contact/>
23. Include IT GmbH. (2015b). *AgenaTrader Andromeda | Tradeescort.com*. Retrieved August 9, 2015, from <http://www.tradeescort.com/en/agenatrader-products/agenatrader-andromeda/>
24. Include IT GmbH. (2016a). *AgenaTrader Wiki | HomePage*. Retrieved March 15, 2016 from <http://www.tradeescort.com/trading-software/tiki-index.php?page=HomePage>
25. Include IT GmbH. (2016b). *Contact Support | Tradeescort.com*. Retrieved March 15, 2016, from <http://www.tradeescort.com/en/support/contact-support/>
26. Muehlen, M., & Ting-Yi Ho, D. (2005). Risk Management in the BPM Lifecycle. In C. Bussler, & A. Haller (Eds.), *Business Process Management Workshops - BPM 2005 International Workshops* (pp. 454-466). Nancy: Springer.
27. Myers, W., & Putnam, L. H. (2003). *Five Core Metrics: The Intelligence Behind Successful Software Management*. s.l.: Dorset House Publishing.
28. Nathaniel, P. (2015). *BPM Everywhere - Internet of Things, Process of Everything*. s.l.: Future Strategies.
29. OMG. (2011). *Business Process Model and Notation (BPMN) - Version 2.0*. s.l.: OMG.
30. OTRS Group. (2015). *Release Notes: OTRS 4 beta1 - otrs.com*. Retrieved August 9, 2015, from <https://www.otrs.com/release-notes-otrs-help-desk-4-beta1/>
31. Paulk, M. (1993). *Capability maturity model for software*. s.l.: John Wiley & Sons, Inc.
32. Ramesh, R. (2007). Your BPM Implementation is Bound to Fail | BPMInstitute.org. Retrieved August 23, 2015, from <http://www.bpminstitute.org/resources/articles/your-bpm-implementation-bound-fail>
33. Ramler, R., & Wolfmaier, K. (2006). Economic perspectives in test automation: balancing automated and manual testing with opportunity cost. *Proceedings of the 2006 international workshop on Automation of software test* (pp. 85-91). s.l.: ACM.
34. Redmine. (2015). *Features - Redmine*. Retrieved August 9, 2015, from <http://www.redmine.org/projects/redmine/wiki/Features>
35. Rosemann, M., & Brocke, J. v. (2010). The Six Core Elements of Business Process Management. In M. Rosemann, & J. v. Brocke, *Handbook on Business Process Management 1* (pp. 107-122). Heidelberg: Springer.
36. Rosenberg, J. (1997). Some misconceptions about lines of code. *Software Metrics Symposium, 1997. Proceedings., Fourth International* (pp. 137-142). s.l.: IEEE.

37. Ruhe, G., & Saliu, M. (2005). The art and science of software release planning. *Software, IEEE*, 22(6), 47-53.
38. Weske, M. (2012). *Business process management: concepts, languages, architectures* (2nd ed.). s.l: Springer Science & Business Media.
39. Whitesides, B. (2016). *What is Agile?* Retrieved March 15, 2016, from <http://www.agileprojectvalue.com/what.html>
40. Williams, L. (2001). Integrating Pair Programming in a Software Development Process. *Software Engineering Education and Training, 2001. Proceedings. 14th Conference on* (pp. 27-36). s.l.:IEEE.