

Sebastiaan Herman
MSc Software Engineering
December, 2007

Rapid application development offshore

A RADICAL NEW APPROACH TO GLOBAL SOFTWARE
DEVELOPMENT

University of Amsterdam
Paydutch



Summary

This thesis describes the creation and validation of “Rapid Application Development Offshore” (RADOS). RADOS aims at increasing performance in offshore development by empowering the development team by applying leadership tactics supported by the latest knowledge on Service oriented architecture. Currently the benefits in offshore software development are decreased by a significant amount of management overhead costs. Empowerment increases autonomy and thus reduces the need for management.

Most approaches to offshore development focus more on the software development process and less on the people that create the software. This results often in excessive specification and process management onsite. Not only are these processes expensive, but they can also make the work of the offshore developers less interesting. Developers translate specifications into working code, with little opportunity to put their own creative thought into the design.

Research in the early 90s has shown that participating leaders perform better than authorian leaders. They have a wider span of control and produce better results; productivity and innovation are significantly higher. These leaders know that a team with highly trained professionals is perfectly capable of making their own decisions and that performance can only be maximized by supporting the team in becoming self-managed. RADOS supports the offshore development team in becoming self-managed and aligned with the goals of the client company.

Service oriented architecture (SOA) is a key ingredient in RADOS. SOA helps stabilizing requirements early in the development process and allows for incremental development by using small releases, manageable by teams varying from one to three developers.

Using SOA, empowerment is possible in offshore development and it can increase performance. Evidence was found that the employees felt generally more responsible for their work; the developers had less problems and more and better ideas about the clients software; towards the end of the research, the developers were generally more productive; Also, developers were directly communicating with important stakeholders in large organizations like the head of ICT from TNT Benelux, the largest shipment company in the Netherlands; and with the support desk of Equens, a major payment cooperation in Europe.

Preface

My interest in moving software development offshore came after a series of bad experiences in hiring qualified personal onsite. I had to wait for months before I could interview a new developer. The quality of their work was low; I interviewed and hired four employees; none of them had formal ICT education. And their price was very high.

It occurred to me that companies in my direct environment preferred paying ten times as much for an onsite developer and waiting months before they come available. I heard complaints about the quality and communication in offshore development; assignments had to be given in extreme detail and after delivery there still was a lot of rework.

I expected the offshore developers to be more motivated than my onsite colleagues, for them this must be an almost unbelievable chance. Knowing this, I started my research in getting to know the people I was dealing with; a process which became the most interesting experience of my professional life.

Many people in my environment contributed to this thesis. One group of people deserve to be mentioned first. My Chinese friends showed me what I already expected; when fighting for the same cause, they can easily match and even outperform the qualities their overpaid colleagues in Europe. What struck me in particular was their capability to acquire knowledge. Little about service oriented architecture, the way of working, and the tools and materials used was known from the start, but in the end the team occurred to me as very professional.

Three of the teams members made extraordinary contributions to this research. 'Jin Bin Lu', who rapidly proved a great help in adopting SOA; 'Dong Hui Chen', who managed to communicate effectively with all sorts of stakeholders directly towards the end of the research period; 'Xiang Bin Chen', who was responsible for the good results of the 'Toolbox 55 project'. Based on the knowledge acquired during this research they started a company named 'Teamwish'. I am looking forward to working with Teamwish.

My special thanks go to my coach Hans Dekkers (UvA). He helped me to grow incredibly fast as a software engineer, team coach and researcher.

A great review was received from 'Bas Terwijn', his review made a big difference to the final result. Other reviewers were: 'Sharif Moeniralm', 'Marco van Gelder', 'Jo Janssen' and 'Marc Hermans'.

Finally, I thank my partner Tatijana van der Veer. During the writing of this thesis I was so busy that the time invested in my role as a partner decreased to almost zero. I admire the way she unselfishly helped me through this process by taking up many of my responsibilities.

TABLE OF CONTENTS

TABLE OF CONTENTS	5
1. INTRODUCTION.....	7
1.1. PROBLEM DEFINITION	8
1.2. POWER TO THE DEVELOPERS.....	8
1.3. CURRENT APPROACHES	10
2. CONTEXT	11
2.1. BACKGROUND VENDOR COMPANY	11
2.2. BACKGROUND CLIENT COMPANY.....	11
3. EMPOWERMENT	13
3.1. THE COGNITIVE MODEL OF EMPOWERMENT	13
3.2. CONTROL SYSTEMS.....	15
4. SERVICE ORIENTED ARCHITECTURE	17
4.1. THE SOA LAYERED VIEW	17
4.2. SPANNING APPLICATION.....	19
5. RESEARCH APPROACH	20
5.1. THE DEVELOPMENT OF RADOS.....	20
5.2. VALIDATION.....	20
6. DEVELOPMENT OF THE METHOD.....	24
6.1. INITIATION	24
6.2. LEADERSHIP.....	25
6.3. SELF MANAGEMENT	26
7. RADOS.....	28
7.1. WORKING PROCESS	28
7.2. WORKPLACE ENVIRONMENT	31
7.3. COACHING	36
7.4. APPLICATION OF THE CONTROL SYSTEMS OF EMPOWERMENT	37
8. MEASUREMENT.....	39
8.1. INSTANT MESSENGER STATISTICAL ANALYSIS	40
8.2. INSTANT MESSENGER EXCERPTS	41

8.3.	RELEASE COMPARISON	43
8.4.	TICKET SYSTEM BACKLOG	44
8.5.	ONE CASE MAGNIFICATION	46
9.	MEASUREMENT RESULT	51
9.1.	DOES RADOS RESULT IN EMPOWERMENT?	51
9.2.	SERVICE ORIENTED ARCHITECTURE AND COMMUNICATION	52
9.3.	RADOS PERFORMANCE	52
10.	CONCLUSION	55
10.1.	RESULTS	55
10.2.	AGILITY IN OFFSHORE DEVELOPMENT.....	56
10.3.	FUTURE WORK	57
	REFERENCES	58

1. INTRODUCTION

Global software development (GSD) is vastly gaining the interest of many companies in the past decades [McDougall, 2006] [Agarwal, 2006] [Minevich, 2005] [Landis, 2005]. In 2006 the size of the business process outsourcing market was estimated \$1.2 trillion according to a research conducted in 2005 by IDC. It grew up from \$300 billion in 2004.

- According to Gartner Inc. and IDC, the market for offshore IT services will more than double from about 3% of overall IT services spending in 2005 to between 6% and 7% of overall spending within the next three years. Gartner expects offshore IT services spending to reach \$50 billion by 2007.
- IDC analysis anticipates that the worldwide IT outsourcing market will grow to \$18 billion by 2008; at an annual compound growth rate of 20%.
- In 2001 GE planned to increase their offshore investment to \$ 400 million.
- In 2006 IBM announced to make a \$ 6 billion dollar investment in India. They hired 43000 Indian workers.

The decision to move processes offshore is typically motivated by cost reduction [Minevich, 2005]. The ACM Job migration task force reported in 2006 that a new software engineer costs \$45,000 annually in the United States and only \$5,000 per year in India [Aspray, 2006]. In the Netherlands, the average cost of a software engineer is currently about \$30,000 annually [Monsterboard].

Lately also the availability of skilled software development employees is becoming an important factor for companies to start moving business processes to offshore locations. In [Minevich, 2005] Expert William Sanford states, "There is a serious issue that the U.S. is not generating enough skilled engineers/technical students to meet internal business demand." This accounts also for the Netherlands, which is number 1 in Europe as it comes to outstanding job applications in ICT at the moment of this research [Automatiseringsgids, 2007]. In India and China there is a high availability of software engineers [Minevich, 2005] [Aspray, 2006].

However, other numbers show that sustainable benefit most likely cannot be achieved using current methodologies. Using them, costs can only be reduced to about 20% [Minevich, 2005] [Fergusson, 2004] [Aspray, 2006] [Boehm, 2003], while the wage difference between a European and Indian software developer varies from 70% to 90%. According to researchers

most of the potential benefit from offshore development is consumed by overhead cost [Boehm, 2003].

India has a leading position in the world market of GSD currently, but researchers are speculating whether this is about to change in the future [Minevich, 2005] [Rhongzu, 2003] [Rajkumar, 2001]. Salaries in India are rapidly rising. As Mercer reported in [Bundy, 2007] the wage of Indian software engineers increased with an annual average of 11.5% over the last 5 years. Until recently China shifted its attention from the hardware market to the software market. In the next 10 to 20 years, the Chinese government will make major investments in ICT [Minevich, 2005]. According to expert Cyrill Eltschinger the average Chinese wage undercuts the average Indian wage by 30%.

1.1. PROBLEM DEFINITION

Currently there is an abundance of highly educated, over ambitious, hard working developers living in another country that ask a fraction of the cost for their labour than their equally skilled colleagues in our own country, and we can hardly benefit from this.

Offshore development projects are often characterized by the plan-driven (or waterfall) approach. In a research under 10 large companies in the Netherlands, Shamsi wrote that 9 of them were using the waterfall method to offshore development [Shamsi, 2007]. Using waterfall, the feedback on a design decision is provided when the software is implemented. The average cost to implement a change in design increases exponentially over time. This methodology proved to be ineffective at onsite development in the 70s [Boehm, 1996].

For the vendor's developers, using waterfall means that their work is downgraded to implementing an already designed product. This likely makes the developers feel dispassionate about the outcome of the task. Also, this provides them limited flexibility to use their creative minds to improve upon the requirements from their own perspective. The developers can only be compliant to the requirements and are not stimulated to validate them against the high level goals of the project.

1.2. POWER TO THE DEVELOPERS

This research was based on the assumption that higher productivity, software quality and cost reduction is possible when software developers are allowed to make more responsible decisions. In literature this is often referred to as empowerment.

With responsibility comes risk. Typically a developer that has a more responsible position can make more decisions without addressing a superior. When communication is hindered by distance and culture, feedback can come very late. Therefore, it is more important that the developer is able to make the right decisions.

As the solution grows, the software becomes more complex. Often at the start of a software development project, every stakeholder has a seemingly clear view on what needs to be build, but as the project progresses it becomes increasingly difficult to manage all the implicit or explicit design decisions made in the past. The software soon becomes a large monolithic structure that no one dares to touch because it lacks explicit rationale. Lack of communication will amplify this problem.

SOA

Recently a new architectural style became popular, which promises to mitigate the risks mentioned. Instead of building a monolithic software system, the solution is build from a network of small autonomous applications called services and is therefore called a Service Oriented Architecture (SOA).

SOA helps in stabilizing requirements in an early stage of the development process because it contains a layer that bundles the software interfaces based on the needs for a specific type of user. During prototyping, this layer can be used to design the high level functionality in close collaboration with the end user. When the requirements are stabilized, the business logic will be created in a lower layer.

Also, the subsystems within the SOA are divided by explicitly defined service boundaries. These boundaries provide a clear context for the developer. Within those boundaries the developer will be able to make better decisions without the uncertainty of affecting their colleague's work, which is also described in [Kroghdahl, 2005].

This vision resulted in the main hypothesis of this research:

Main hypothesis: The methodology RADOS empowers offshore software developers resulting in increased performance

1.3. CURRENT APPROACHES

CAPABILITY MATURITY MODEL INTEGRATED

In global software development CMMi is used in a setting that is much related to the rational unified process. It is iterative by nature; it defines similar processes; it relies on phased development.

According to research conducted in 2003 [Goldenson, 2003], benefits in terms of cost reduction, predictability and quality were significant. Improvement rates in all dimensions were varying between 10% and 20%.

However, research also points out that adopting a software process like CMMi alone is only a part of the measures needed to become successful in GSD [Bhat, 2006] [Boehm, 2003] [Damian, 2003]. Bath identified the lack of shared responsibility as the most intricate problem in global requirements engineering [Bhat, 2006].

AGILE METHODOLOGIES

Approaches closer related to RADOS are the Agile methodologies; these focus less on processes [Christiansen, 2006]. Agile methodologies concentrate more on vision and values which motivate the team to better work with the customer. Frequent and face to face contact with the customer is an important part of agile methodologies.

In GSD, face to face contact with the customer becomes a problem that is not easy to be solved. Different time zones decrease the number of hours to get into direct contact; video conferencing can be a problem if bandwidth is too low; geographical distances make face to face contact less cost effective [Christiansen, 2006].

2. CONTEXT

This research was based on the development of an online payment service provider solution by Asia SD (this name is fictive), a vendor in china for Paydutch a company in the Netherlands. The following paragraphs provide background information of the vendor and client companies. The last paragraph provides information on three distinct research periods.

2.1. BACKGROUND VENDOR COMPANY

The vendor company is named “Asia SD”. It was founded in 2002 by a Chinese businessman who migrated to the Netherlands in 1999. It started as a company with one employee, the CEO. Within the 5 years of its existence it has grown to 20 employees.

At the start of the relation with the client, the vendor had no stable software process. Most projects were small and done by a single employee. Awareness of quality and focus on the customer were generally low.

The first project that the client used as a trial project was calculated fixed price. The project was expected to take 2 months. However, after experiencing a delay of 1 year, the project was terminated by the Vendor; the project could simply not be finished.

2.2. BACKGROUND CLIENT COMPANY

The clients company started in 2005 as an online escrow service provider to secure consumer to consumer transactions online. An escrow service operates as a trusted third party for safekeeping money in transactions involving large purchases. Paydutch offers this service also for smaller purchases between consumers. The consumer to consumer (C2C) market is rapidly growing. The internet makes it easier to buy and sell used goods. However, it is also commonly used by fraudulent merchants.

In 2005 however the Dutch market was not ready for escrow yet. Many showed interest, but few started using the services offered. The management concluded that PayDutch needed a startup partner that could generate a lot of transactions.

Most of the potential startup partners were interested in services different from those offered to the C2C market. The basic workflow was forcefully adapted in many ways. Eventually it

became clear that the architecture that was based on providing escrow services was not flexible enough to support these changes.

The project manager had a strong believe that GSD could help the company innovate faster. The price paid for an experienced developer in China was only a small part of the price paid in the Netherlands. Chinese developers were thought to be more collaborative because the collective nature and the availability of developers is higher in China.

COMPANY VISION

At the start of the project PayDutch was an escrow service provider. However, during the projects life cycle, the central vision changed. The focus from escrow service provider shifted to that of a payment service provider. This change process caused instability for the project and made the employees feel ambiguous.

It took 5 months for the project leader to capture the vision with two metaphors that described the company and its related products. The company was referred to as a payment innovation platform and the products created by the company were referred to as payment adapters, because the products adapt payment services to the needs of the customers. These helped in recapturing a stable vision and focus of the employees.

3. EMPOWERMENT

Thomas & Velthouse based the definition of empowerment on the employee's ability to assess a task positively in four cognitive dimensions [Thomas & Velthouse, 1990]. A task is a unit of work that is assigned to or selected by the employee that executes it. The assessment is done before the task is executed, and will determine the behaviour of the employee while the task is executed.

Empowerment involves bringing power down in the organization to increase innovation agility and decrease the need for managers. Practices focus on increasing employee confidence and feeling of self-efficacy. The role of the manager in this process changes from being an authoritarian leader to being an employee coach that creates an environment that fosters innovation and productivity [Thomas & Velthouse, 1990]. Several studies show that empowerment results in higher productivity and innovation [Spreitzer, 1995] [Spreitzer, 1996] [Simons, 1995] [Thomas & Velthouse, 1990]; it also reduces the need for control [Simons, 1995] [Thomas & Velthouse, 1990].

3.1. THE COGNITIVE MODEL OF EMPOWERMENT

In figure 1 is depicted the cognitive model of empowerment as was proposed by Thomas&Velthouse in [Thomas & Velthouse, 1990]. When an employee needs to do a certain task, he or she always does a mental assessment of the task in these 4 dimensions:

- **Competence:** Am I able to execute this task with an agreeable amount of effort?
- **Impact:** Does this task have significant impact on the business goals?
- **Meaning:** Is this task important to me? (this is based on the personal values of the employee)
- **Choice:** Will I have significant influence in the outcome of this task?

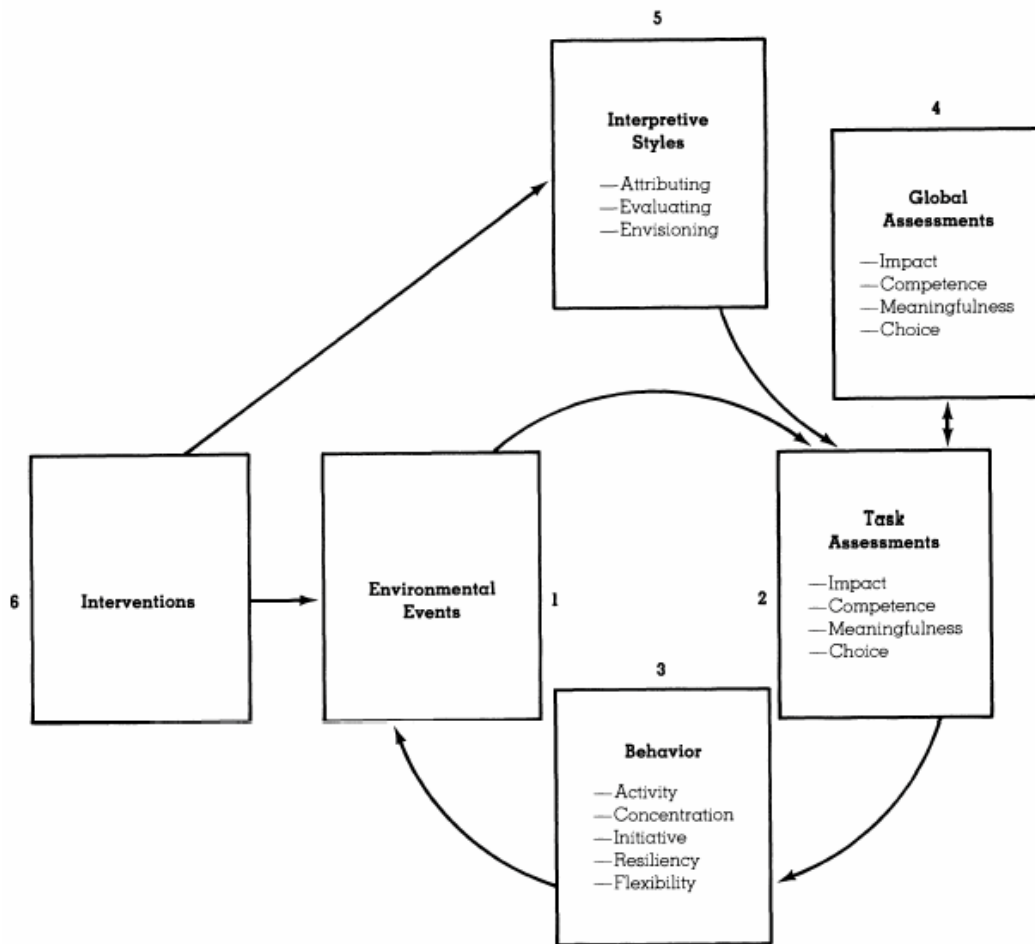


Figure 1 Cognitive model of empowerment

Based on this task assessment the employee will express a certain behaviour.

Empowerment in literature has been linked to an increase in: activity, concentration, initiative, resiliency and flexibility [Thomas & Velthouse, 1990].

The global assessment (fig 1) is the incremental result of past task assessments. This will have either a positive or negative influence on the employee's ability to assess the task with a positive outcome in all 4 dimensions.

This model assumes there are two ways to influence the employee's task assessment and thus the employee's behaviour: Through deliberate interventions in the employee's workplace environment; and by providing feedback on the employee's interpretive style, which is used in psychology to make a person aware of its style when it is ineffective. One example is that a person assessing a task can blame either the task or itself when not being able to execute the task. These styles contain respectively internal and external attribution (fig 1.5).

3.2. CONTROL SYSTEMS

In [Simons, 1995] Simons described 4 systems that help in getting control over empowerment. Empowerment can be a blessing, but it can also be a beast. As employees become more autonomous, can make more decisions by themselves, temptation lures to misuse this freedom. Kidder, Peabody and company lost \$350 million when a trader allegedly booked fictitious profits; Sears, Roebuck and company took a \$60 million charge against earnings after admitting that it recommended unnecessary repairs; Standard Chartered Bank was banned from trading on the Hong Kong stock market after being implicated in an improper share support scheme. Therefore, levers are needed to keep empowerment in line with the company perspective.

DIAGNOSTIC CONTROL SYSTEMS

Provide feedback information on performance problems. It can be seen as the dials in the cockpit of an airplane to scan for signs of abnormal functioning. Whenever a red light pops up, it is an indication for further research. Managers use this to track progress towards preset standards of performance, or company goals. Feedback can be used to fine-tune inputs and processes so that future outputs more closely match goals. This feedback should solely be used as an indication for further research and should not be used to ensure effective control. In fact, doing so can be pressurizing and counter-effective. For instance, using diagnostic control systems to hold developers directly accountable for the lines of code they produce per hour will likely produce low quality code.

BELIEF SYSTEMS

Make the employees understand that they can contribute to the company's goals. Belief systems are used to communicate the company's core values and mission. It shows workers how the company makes value, and gives a better perception on their unique contributions to this process. The need to contribute is inherent to our existence, but companies often make it really difficult to see contributions related to the higher goal. Making this relation clear therefore gives the worker a higher feeling of self-efficacy. For instance, a software architect that deeply understands the projects goals and is able to make decisions based on these goals intuitively shall likely be able to create an architecture that will be longer lasting.

BOUNDARY SYSTEMS

Boundaries should only tell a worker what not to do, instead of what to do. It will prevent them in making incorrect decisions because of mistakes, pressurizing circumstances or temptation. It protects the company from potential harm, and will provide workers with more

energy to decisive power, because knowing that you are at least not doing the wrong things will provide self confidence and thus help making more creative decisions.

INTERACTIVE CONTROL SYSTEMS

Small organizations have the advantage that senior managers have face to face contact with their subordinates, but as organizations grow larger, this becomes increasingly difficult. Interactive control systems are the formal systems that allow senior managers in large organizations to influence the decisions made by their subordinates. The data that flows from these systems are of strategic interests, and need to be discussed in face to face contact with senior managers. Furthermore, it is a catalyst for an ongoing debate about the underlying data, assumptions, and action plans.

4. SERVICE ORIENTED ARCHITECTURE

SOA supports alignment between the business processes and the IT infrastructure on a higher level of abstraction [Zimmerman, 2005]. Levi writes that SOA is a powerful lever for strategy [Levi, 2002].

Service oriented architecture can be used to improve scoping by incrementally refine the solution from the stakeholders point of view. In addition to traditional architecture principles like information hiding, modularization and separation of concerns, SOA provides service composition. This means that the user's functionality is composed in a dedicated layer instead of in the user interface [Levi, 2002] [Zimmerman, 2005].

4.1. THE SOA LAYERED VIEW

Figure 2 depicts a simplification of the layered view of PayDutch. The layer on top describes the physical business choreography in the form of use cases. The layer on the bottom represents external operational systems, for example the database server or external service providers like the SMS service. The service coordination layer delegates requests initiated by business activities to the service provider components. The service provider layer contains business logic and adapters that adapt data so that it conforms to the expectations of the external operational systems; we have no, or only limited, control over these systems expectations.

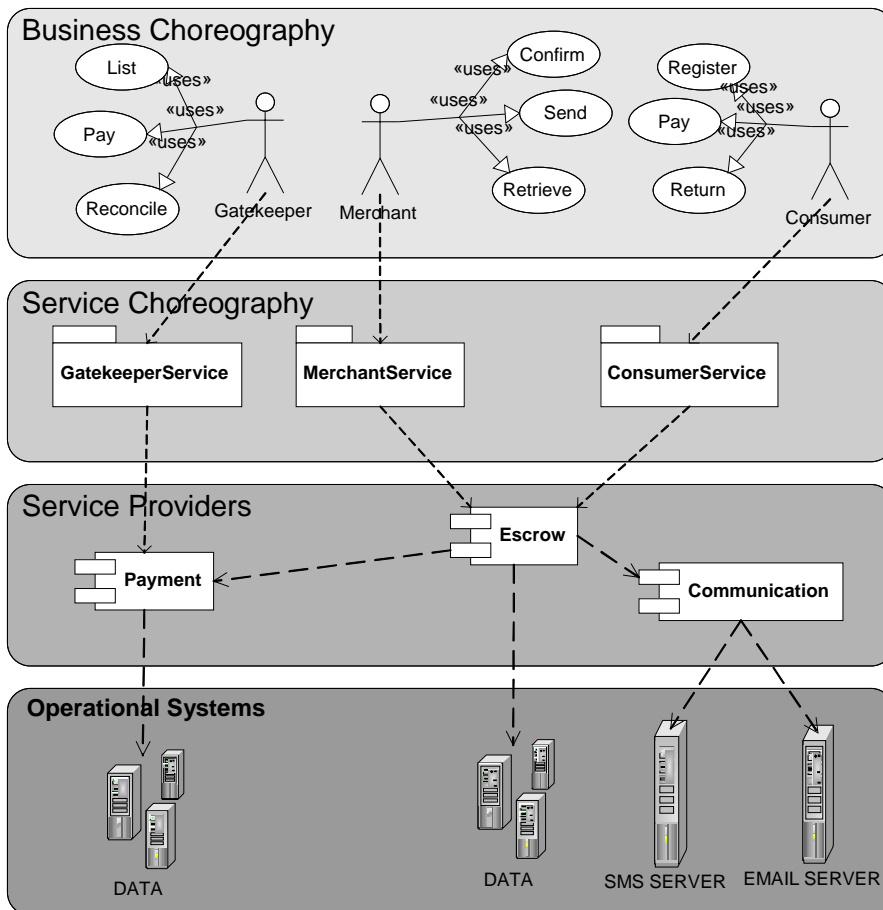


Figure 2 PayDutch architecture

The primary driver of software development in RADOS is the use case diagram. The use cases (Business orchestrations) are implemented in the service orchestration layer. The use cases and actors are mapped onto physical code, causing the abstractions used by business owners to correspond with the abstractions used by developers. Each service represents a user role and each method represents a use case in the service.

Working in similar abstractions as the business owners supports communication. For the developer this provides a better perception on the meaning of a task with respect to the agreements made with the business owner. The business owner is better able to explain what the component means for the company.

4.2. SPANNING APPLICATION

Prioritizing requirements of a software application is most clear after implementation of the product. However, prioritizing development should be done as early as possible in order to get an accurate overview of the budget and time needed for the project.

SOA allows easy development of a spanning application, using a technique called Breadth first service development. A spanning application is the minimal approach to providing stakeholders feedback on the total solution. Breadth first service development techniques focus on the creation of the totality of services without implementing business logic or persistency; invoking the interfaces will provide only hard coded values [Woolf, 2005]. These services can be attached to a light weight user interface, which provides the stakeholders an overview of the total product.

5. RESEARCH APPROACH

To find evidence for the hypothesis, the following questions need to be answered:

1. How to construct RADOS?
2. Does service oriented architecture support communication?
3. Does RADOS result in empowerment?
4. Does RADOS outperform traditional offshore development methods?

The first question will be answered by the following paragraph. The second question will be answered by evaluating existing literature on SOA. The third and the fourth question need to be answered with factual evidence.

5.1. THE DEVELOPMENT OF RADOS

In the first setup of RADOS, the vision was used in the creation of an IT infrastructure; an initial architectural concept and the hiring of one offshore employee.

From that moment the methodology was iteratively improved. An iteration started with a measure that addressed problems found in the period before that change. After the change was implemented the result was evaluated after two to four weeks. Each iteration the methodology became more concrete. This process is described in detail in chapter 6.

5.2. VALIDATION

The RADOS methodology was validated by finding answers to the research questions. Validating a method is hard. This chapter describes the challenges in validation. It also describes the data sources used in the validation.

PROBLEMS IN DATA VALIDATION

The methodology was created and tested on a real project with real stakeholders and real concerns. This meant that a lot of factors were not under control which made validation hard.

Moving target: It became difficult to clearly define the project enabling comparison to a similar case. During development, the vision of the business made a dramatic change. With the vision, also goals and the requirements change. Changes like these have severe influence on the projects time to market.

Cause and effect: A positive change in behaviour and productivity cannot easily be traced back to its root cause. A multitude of environmental elements can be the basis for the effect, or it can be only one. Also, we do not have the luxury that we can create two identical worlds and where we can measure the difference that a decision makes when all other variables are equal.

Personal involvement: The author of this thesis was involved in the project as the manager. Having two roles made it hard to make an objective measure. The manager had a responsibility towards its colleagues to lead the project well, and the author had a responsibility as a researcher to objectively detect and investigate problems. During the writing of this thesis it caused the author to continuously switch roles. It is hard to identify a problem without stepping into the manager's role and trying to solve it.

Disruptions: The project involved many stakeholders, from different companies. Aligning a process between so many stakeholders increases the number of external disruptions. These disruptions make it harder to measure performance. For example there was a lot of delay in waiting for authorization on external systems. Also it was often hard to get the feedback in time from an external party.

Product metrics: Code quality metrics like coupling and cohesion were not measured. SOA allowed us to have proper control over these aspects on an architectural level.

Cognitions: Empowerment is a cognitive state of mind. It is not easy to measure what goes on in someone's mind. If that person lives in another country, speaks another language, this becomes even more difficult.

Hard to compare: Innovation is difficult to measure, and therefore difficult to compare. It is largely based on the capacity of the ones involved to assess a situation and make a decision that supports the business goals. Innovation always involves making new products; the rate of innovation is the speed of producing new products and bringing them on the market. However, it takes time to bring a product on the market, and also there is more to making a product successful on the market than innovation.

DATA VALIDATION

The validation of the hypothesis was conducted using three sources of data. These are explained in the following sub-paragraphs.

Instant messenger excerpts: During the whole research period most of the communication with the vendor's employees was conducted via an instant messenger service (IM). The IM history provided insights in behaviour that could be linked to empowerment. Communication via IM is more informal than pre-defined tickets or created issues in the systems.

IM was the primary source of information on the behaviour that the vendor's employees expressed towards the coach. During the research period, almost all of the direct communication went through IM. These conversations were all logged. The logs contained behavioural information about employees, because the medium induces an informal atmosphere.

First, to find evidence for the presence of empowerment, we did a pre-selection of excerpts that could contain behavioural information that relates to empowerment. Second, we examined the selected excerpts using an accurate definition of empowerment. Third, we searched the excerpts for evidence that could possibly falsify the prior arguments.

Instant messenger random data: Some of the information regarding empowerment could be found using evenly picked conversations from the first 1000 lines and the last 1000 lines of IM history. This more or less corresponds to the first and the last month of the research period. The reason for randomly picking conversations from the IM history was that the amount of data was too large to examine.

Release comparison: Within the research period there were two releases which could be compared in order to find the effect of implementing RADOS. This data provides evidence for the impact of RADOS on productivity.

Ticket system backlog: For most of the formal communication regarding the project a collaboration system was used. This system used tickets to communicate project related tasks. We used the data that came from this comparison as part of the evidence that supported an increase in innovation, which results from empowerment [Thomas & Velthouse, 1990].

One case magnification: In the data sources we identified one case where we could overview the effect on empowerment in one particular assignment. We found a spectacular

increase in code check-ins. We found evidence in IM history that empowered behaviour could be related to this increase.

We used this data to link an increase in performance to empowerment and RADOS.

Measurement result: In the measurement result we discuss the data with respect to empowerment, performance and the relation with RADOS.

6. DEVELOPMENT OF THE METHOD

The project started in January 2007 and ended in September 2007. During this 9 month period the project manager overcame some serious problems. The research was divided in three periods.

In the subparagraphs below the periods are described by a summary of what happened, following the effect it had on the methodology. The latter is described using a grey box.

6.1. INITIATION

January – March

In this period we structured the software development team. In 3 months we hired a total of 5 employees, one of them was hired on site.

Knowledge on service oriented architecture needed to be developed on site and offshore; meanwhile the project was already under development. At first a prototype was build using persistent data, but later it became clear that mocking data on the service interfaces was a more effective way to rapidly mimic the functionality needed to capture new requirements.

Project management became difficult. We received a lot of questions from all the employees about the software that needed to be implemented and the lack of overview made it difficult to track progress.

Impact on RADOS design

Software development statistics

Statistical tools were implemented that helped in detecting problems by providing feedback on the work.

Collaboration platform

A collaboration platform is an online tool that supports the collaboration of graphically dispersed teams. For software development it often contains a wiki and an issue tracker. The software is preferably reachable via the web.

The issue tracker was used from the start, but the wiki became increasingly useful during the first period. The nature of communicate messages was mainly functional like 'How to configure your development machine' for new developers.

Employee motivation

The coaches' focus shifted towards the importance of employee motivation. The coach started listening to the individuals to find a fit between their goals in life and the company goals.

6.2. LEADERSHIP

April - June

During this period there were many rotations in the team leadership position at the offshore vendor. Due to the rapid hiring of new employees, the work for the client's manager soon became too time-consuming. The manager decided to put one of the offshore developers in the position of a leader. After that 2 more leaders were assigned, but all of them failed to manage the project correctly.

The task of the first offshore team leader was to create and maintain tickets in the issue tracker. The client supported in the creation of those tickets and starting tutoring the offshore developers in writing quality code and in experiencing the needs of the customer.

The first leader was technically very capable. However after one month it turned out he not only created all assignments, but also executed most of them himself. He found it difficult to give assignments to other employees. The result was that his colleagues had almost nothing to do, while he was working overtime.

The second leader was a better communicator by nature. He was very capable of collaborating and evenly distributing the workload. However, after a short period of time it became clear that his technical capabilities could not provide him the insight needed to create good tasks and oversee the impact of his decisions.

The third leader was a more experienced leader. He already led a large team at his former company. He was technically capable and was also a fairly good communicator to his colleagues. However, during a migration project there were misunderstandings between the

client and that leader. This led to unnecessary reimplementation of several large parts of the project, which resulted in a month of rework by estimate.

At the end of the period we had to conclude that leadership was too error prone. First, there is little overview on how a leader emotionally performs in his position. We are not able to intervene on problems like these within a timely fashion. Second, miscommunication with a leader is a far greater problem than miscommunication with one employee. The result can be a large amount of incorrect functionality.

Impact on RADOS design
<p>Multiple small releases</p> <p>This allowed appointing an owner to each release who bears the responsibility of that releases successful implementation.</p>
<p>The personal weekly report</p> <p>To reduce the need for management, the personal weekly report created a sense of responsibility towards each other. Individual contributions became explicit, as well as motivational problems.</p>
<p>No single appointed leader</p> <p>Leadership turned out to be a liability. The impact of miscommunication became too large; and it was difficult to oversee which employee was really able to bear the responsibilities of being a leader.</p>

6.3. SELF MANAGEMENT

July - September

Leadership was taken back by the client company. The project was divided in several small projects. Small teams were identified containing about 1 to 3 employees. Also a weekly personal report was institutionalized, to create a better sense of responsibility at the employee's side and a better overview of the work that is done each week.

The client manager changed its focus from managing the project to coaching the developers to manage themselves. The project was cut into several small releases. All employees

agreed write a weekly report about their work for that week and the planned work for the next week.

Impact on RADOS design
Self-management
The attention shifted from communicating project related information towards vision and value related information. This resulted in a better sense of sharing a goal with the vendor

7. RADOS

RADOS empowers the developers at the off shore location by providing training, problem ownership, team building and coaching.

RADOS motivates the team to execute the task in line with the business goals; instead of only following up requirements. Deeply understanding the project goals and vision will help the team in understanding requirements validity. By working with high level requirements the developers have a task to further specify and improve requirements. This will enhance understanding and bring more brainpower into problem solving.

Training focuses not only on competence but also on principles and fundamentals. By sharing rationale the team is stimulated to contribute to better decisions and feel responsible for attaining the business goals. According to Thomas & Velthouse [Thomas & Velthouse, 1990], this will lead to more initiative.

The workplace provides an environment that positively contributes to the employee's ability to assess a task positively. For instance, the perception of doing meaningful work can be improved by providing information that reflects the relation of a specific task with respect to a shared vision.

For example, we have the vision of an architecture that is optimized for innovations related to payment. The creation of a public subscribe framework for the payment service provided a significant contribution toward the realization of our shared vision; it made implementation of the component more flexible.

The following paragraphs describe the constructs of RADOS.

7.1. WORKING PROCESS

The working process is based on the Rapid application development (RAD) process. Rapid application development (RAD) is an iterative software development method that builds software incrementally by bringing business people and technical people together. Each increment in RAD is called a time box. The only variable is the functionality implemented, which is used to overcome problems with delay in software so that development better integrates with the business process [Beynon-Davies, 1999]. Also, technical people often are

faced with many business related problems. Bringing business people together with technical people dramatically shortens the project time according to research [Wood, 1995].

The phases in RADOS are intentionally kept small so that it provides flexibility; it decreases the employee's feelings of ambiguity towards its role within the working process, without decreasing the room for creativity in executing the task.

Ambiguous feelings are negatively related to empowerment in literature [Spreitzer, 1996]. However, creating too much clarity will likely to compromise room for bottom up improvement of the development process, which in turn is likely to increase the employees feeling of self-efficacy.

There are three phases in RADOS (fig 3). Each phase has one entry, one re-entry and one exit event. Respectively: the signoff event, the deliver event and the review event. After the signoff of the deliverables a post mortem report is created by the software development team. This report is a functional assessment and used for capturing and institutionalizing the lessons learned.

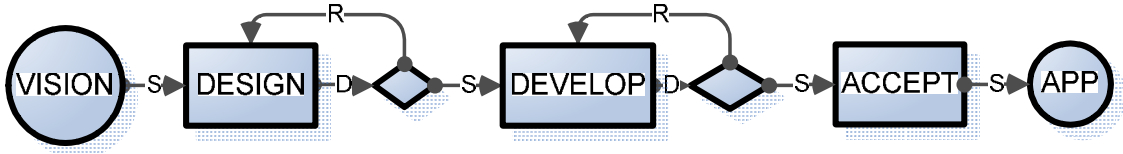


Figure 3 RADOS working process

In table 1 an overview is provided of the intent and deliverables of each phase in the working process. The first row shows the intent of the phase, the second the deliverable that is produced and the bottom row show how new requirements are treated. In the design phase, all requirements are immediately captured in the design to generate rapid feedback. In the develop phase, requirements can be implemented, but they need to be formally approved. In the accept phase has the most strict selective procedure, only high priority requirements that are approved by the project manager should be implemented in this phase.

Table 1 Phases in rados

	Design	Develop	Accept
Intent	Stabilizing req.	Implementing req.	Integrating application
Deliverable	Req. document Design documents Spanning application	Solution Acceptation test Manual	Integrated solution
Implement	All requirements	Approved Req.	Approved Req. High Prio.

In the following paragraphs each phase in the working process is explained. Each paragraph will contain a description of a phase, following an explanation of the incoming and outgoing events and the deliverables that will be produced. An exception will be the vision and the application phase; these phases lack respectively the incoming and the outgoing event.

VISION

Instead of an elaborate requirements document, the client needs to focus on the high level project goals, problem description and vision behind the project. A clear and compact vision will allow developers to better understand the clients point of view. Sharing visions is also an important part of the belief system as was described by Simons [Simons, 1996].

DESIGN

The vision document will be translated in design documentation and a spanning application, which are used to elicit requirements. The design will mostly be done by the vendor, which makes them owner of the problems that come with their design choices, and so it is likely to relieve the client of many design related questions.

The service orchestration layer in SOA will be used to create a spanning application. The spanning application helps in prioritizing the functionality and estimate the total effort needed to implement it. It is a minimal version of the total application containing all the functionality, but no business logic. For example, the spanning application of the escrow release was comprehensive in functionality, but the data in the forms were not stored in a database and there was no data validation.

This phase is intended to stabilize most of the requirements. Changing the specification will be a more informal process. The design is subjected to an open discussion, and requirements change rapidly.

DEVELOP

The spanning application will evolve into the final solution. Service providers can be created concurrently because the most important requirements became stable in the design phase. Implementing requirements in this phase is more costly and therefore additions need to be formally approved before they will be implemented.

It is unavoidable that the data contracts evolve during this phase; in most development projects it is impossible to stabilize all input data upfront. However, these changes won't affect concurrent development. The service providers that are closest to the service coordination layer are built first; as long as the definition of the service does not change concurrent component developers won't bother each other during development.

After the offshore team verified that all requirements are implemented, the software will be delivered to the client.

ACCEPT

During this phase the software will be accepted by the stakeholders. It will be installed in the acceptance test environment. This environment is as real as possible without interfering with live data and processes. All stakeholders will be invited for a meeting where the working of the application will be explained, as well as their part in the adoption process. Requirements changes will occur. However, those are likely to be of low impact.

7.2. WORKPLACE ENVIRONMENT

RADOS includes the following: An online collaboration system; personal weekly reports; automated builds; and the coach.

ONLINE COLLABORATION SYSTEM

The collaboration platform is the basis of all project related communication. It is an easy access to information about the company, the project and related practices. It generally takes the form of a website as shown in figure 4.



Welcome to the Paydutch Wiki

Paydutch is an online escrow system. Its goal is to make online trading 100% safe. Currently on the one site we have customers piling up. On the other site we have a team with unique qualities. The mission is to satisfy the customers with a payment service provider application that is so flexible that we can serve virtually all needs that involve online payment. No matter the form or moment of Payment requested, we can provide it.

Toolbox 55 is ready to go live

Yes
No (**donghui, sebastiaan, Gino**)

Company::

- ⇒ [Our Vision](#)
- ⇒ [Our Boundaries](#)
- ⇒ [Working Process](#)
- ⇒ [Ticket per department](#)

Teams::

- ⇒ [Yardi Block: 1](#)
- ⇒ [Teamwish Block: 0](#)
- ⇒ [Heerlen Block: 2](#)
- ⇒ [Haarlem Block: 1](#)

Figure 4 The wiki home page

The wiki is used to communicate with all departments of the company. It provides access to company information like: the software architecture, the vision and values and project related information like the issue tracker. It also contains a news section. News provides information on upcoming projects and feedback on the result of projects previously developed.

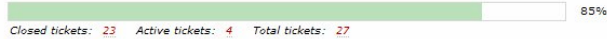
Access to company related information that is up-to-date is an important ancestor to empowerment. It helps in linking tasks to the overarching company goals and thus in correctly perceiving the impact and meaning of a task [Thomas & Velthouse, 1990]. Also sharing this information with the offshore development team is a way of showing that you care. The offshore developers will feel more involved.

For each team, there is a separate page in the wiki. It contains a description of the team and its responsibilities, there are photos of its members and it contains the active tickets. This enables easier identification of the work role. The team pages provide a better idea of the role of the team in the company.

The releases are shown in the collaboration platform as milestones. It shows the vision that preceded the release; it shows the progress in terms of open and closed tickets. This progress is subdivided per team. Also it provides a link to the release page which contains more elaborate information about the release, like: designs documents, test plans and team members.

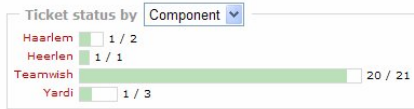
Milestone The Toolbox 55 Release

Due in 5 days (09/21/07)



Vision -> Design -> **Develop** -> Accept -> Live

Toolbox 55 is a company that facilitates work assignments for short term applications. The employer can place an application in Paydutch and Pay the amount of money that stands for it, which can be accepted by an employee. Paydutch will facilitate payment. After the work is finished, the employee can log into toolbox 55 to indicate the work is done. The employer has 7 days to notify any dissatisfaction about the work. After those days, the employee will get the money from [PayDutch??](#)



Team members: [Click here](#)

- 9/21/2007: Delivery date.
- 8/3/2007: Official presentation of prototype to stakeholders.
- 7/6/2007: Use case designs, Role descriptions, screen designs ready to be reviewed

Figure 5 Milestone view

The issue tracking system provides support for division of the work in small releases. The active releases are contained in a roadmap. Each release has a page which contains information about: important dates; active workers; active tickets; tickets that block the working process and design documentation.

To make the wiki an effective tool for communication, the employee’s attention needs to be drawn to this platform. The issue tracking system is contained in the wiki; which is continuously used. Also we keep the data on the issue tracker alive, by adding a news section that information from throughout the whole company.

Also, there are regular polls on the issue tracking system, which allows for all party’s to vote on a subject that matters to them. The result provides information on what goes on in the hearts and minds of all team members of different teams. This can be used to alter the approach. Furthermore, the employees have the opportunity to express themselves, which creates the feeling of being understood.

AUTOMATED BUILDS

Every time an employee checks in code in the version control repository, all the code will be rebuilt and unit tested with NUnit. Also NCover is run to measure the number of lines that are

executed during the unit tests. This provides trust in the central code base. The developer will have direct feedback on the quality of his code.

A defective build can produce large overhead costs. This is even worse in GSD. The lack of direct contact can cause developers to be searching for bugs in code that was build by another team.

A failing build should be fixed as soon as possible, by the developer who made the bug. This is one of the few rules RADOS has. Fixing a bug is relatively difficult if you did not create the bug yourself.

WEEKLY REPORT

To stimulate the group process all team members will write a weekly report of their efforts the past week, and their planned efforts in the next week. If the team members become self managed, they will be mostly deciding their own actions and tasks. With this freedom an employee can easily lose focus of the overall process and misperceive the priority of their task. Writing a report about your contribution to the project helps in projecting your work in the greater context of the project. This report is assembled per team and then made accessible throughout the whole company.

A side effect is that a personal weekly report positively influences motivation. Because the weekly report is made public it is nice to be able to write something interesting in it. Being able to do so, makes you feel competent about your accomplishments. Not being able to do so means you lack contribution to the group process. It will likely motivate to put in some extra effort the next week.

Also, the weekly report creates shared responsibility and accommodates learning / reflection. It allows writing about the week, and the hurdles one needed to overcome to or still are open that make the work difficult. Hence, it provides a little insight in the other team's workspace and thus is good for understanding.

Weekly reports that show little progress are often a sign of motivational problems. It is an indication for further research on motivational aspects that cause the employee to stagnate.

Besides providing regular feedback, the weekly report is an effective means of creating a shared culture among the team members working on different locations. It is a source of

information about the successes and setbacks other teams are experiencing by working for the same benefit.

TIMELINE

The timeline (Fig. 6) is part of the diagnostic control system. It provides feedback on project related activities as: code, ticket and wiki changes. The timeline provides easy access to those changes, which facilitates reviewing work and provides feedback on progress. The timeline provides feedback on day to day activity. It contains: wiki changes, code check-ins and ticket changes.

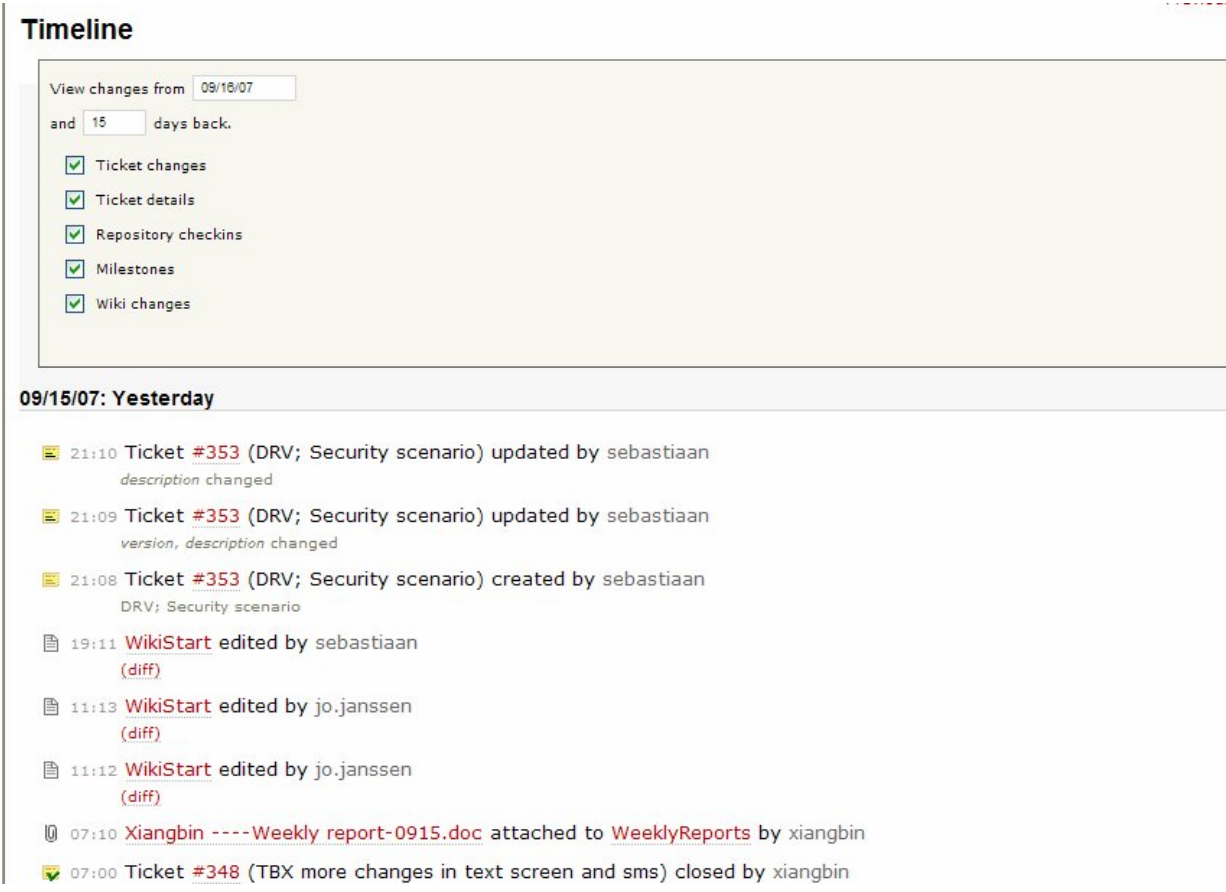


Figure 6 Timeline

Every code check-in is logged on the page with a link to the repository. Clicking this link shows the changes in code. It is an easy way to do an in depth checks on the code that is produced by the team.

Although this looks like 'Big Brother' in the first place, but in practice this is not the case. The coach of PayDutch received a lot of explicit request to check upon code because the employees were very willing to improve their quality standards.

As with the weekly report, those employees who are not able to check-in code on a regular basis are more likely to have motivational problems. For the coach this means he needs to try and figure out what the real problem is.

7.3. COACHING

The coach in RADOS is a trusted source of feedback on performance. He gives credit to the individuals while being on the background himself.

SHARING VISION AND VALUES

The coach also articulates the strengths and weaknesses of both the client and vendors culture; of the company as well as the country. The basic message of each culture is always available in a document, but the coach will keep it alive and will relate actions and events to the cultures.

Sharing culture between partnering companies in global software development is very important. As described in the case study of Mao [Mao, 2007], for a vendor in china and a client in Japan, sharing culture was a very important part of their claimed successes. Another researcher Bath also identified sharing culture as an important strategic lever.

A starting point for controlling motivation is finding individual desires. Because the coach is stimulating individual growth, the employees become receptive for his or her information. This makes the coach a good carrier for company related, vision and value loaded messages.

As soon as the needs of the individual's surface, the coach will find ways that allow the employee to accomplish these goals in a mutual beneficial way. Aligning the needs of the employee with the company goals will make the employee feel more involved.

SHARING RESPONSIBILITY

If the coach detects a problem, he will try to make clear how this problem manifests in the running application. For example, when there is potentially harmful code checked in, it is good to create a real world scenario that describes the impact of a failure by error prone

code. The objective as a coach is equipping the employee with the background information that is needed to solve the problem rather than solve the problem himself.

Having the knowledge to solve a particular problem may tempt you to solve a problem for an employee without providing rationale. This could solve the problem, but possible better alternative viewpoints remain unspoken. Also it is likely to increase the dependency of the employee towards you as a problem solver. You are better of explaining the employee what the important facets are in solving the problem, and thus equipping the employee with deeper knowledge instead of only the answer to a problem.

Here follows a good example from MSN history:

Jinbin:

When the merchant have not joined the escrow in a long time? The transaction will Expire), and PayDutch will need to return the payment to the consumer. Does PayDutch calculate the fee?

Sebastiaan:

On the one side you can state, we reduced the risk for the customer that all his money is gone. Normally without PayDutch, it would be. But on the other side, a customer is more likely to use PayDutch when he get all his money back after he used the application, when something went wrong.

In this example the coach provides the employee with information that allowed him to make a judgement from its own point-of-view. He explained to the coach that similar services are already active in China and that they always return the money without calculating a fee when the escrow deal is cancelled. The result was a better solution than we could provide because by not giving the answer directly, we drilled important knowledge that would otherwise probably remained tacit.

7.4. APPLICATION OF THE CONTROL SYSTEMS OF EMPOWERMENT

As empowerment is difficult to control onsite, onshore this will be even more difficult. There are differences in culture and working mentality and the vast distance is also troubling communication. The autonomous empowered employee will therefore have more difficulty to understand the company course and will be less likely to feel part of the company. This will

also likely increase the temptation to use the freedom provided to do things only for individual purposes, which could harm the company in the end.

Table 2 describes how the control systems of Simons [Simons, 1990] are implemented to prevent this from happening.

Table 2 Implementation of the belief systems

	Belief system	Boundary system	Diagnostic control
Company vision	V		
Coaching	V	V	
Company rules		V	
Working process		V	
Timeline			V
Weekly report			V

The company vision will be available using the collaboration system, and articulated and provided with contextual meaning by the coach. Also the coach is responsible to distribute ownership of the vision, value and company goals amongst the employees.

The working process and some company baseline rules are communicated using the collaboration system. This information is very clear, very basic and therefore easy to understand and live up to. The working process explains the nature of activities within a certain stage of development. For example, the developer will not be tempted to change requirements without formal approval of the project manager after the design phase if this regulation is communicated very clear.

The weekly report reflects the employees activities. This information is useful for the coach to see how the team is performing and whether a member has problems in finding useful tasks.

8. MEASUREMENT

During a research period of 9 months, four Chinese offshore developers produced 250k lines of code, of which approximately 40% was generated. The project was lead by the author located in the Netherlands. During the research, two projects were delivered; the main architecture with the basic services and one extension that facilitated in labour service transactions. Besides that, three new projects were started, two of them were aborted because of changed priorities and another one was finished in December; which was developed in a partnership with TNT postal services in Belgium.

Empowerment was captured using behavioural analysis of the instant messenger history, which was the dominating channel for informal communication. Because empowerment is a cognitive state, measurement could only be done by relating behaviour to its likely psychological origin. The excerpts used for this research can be found in the appendices at the end of this document.

Another element in measurement was the presence of increased performance, originating from empowered employees. Performance was measured using analysis of the code and issue tracking backlog. The relation with empowerment was found by analyzing the existence of known behaviour linked to empowerment and performance indicators from subversion statistics.

In the table 3 an overview is created of the measurements and their intent. The first column contains the four cognitions of empowerment, then the expected resulting behaviour which is beneficial to performance [Thomas & Velthouse, 1990] (and therefore categorized under performance). Innovation is measured as the number of approved ideas expressed by the offshore developers in instant messenger conversations, which can be regarded as an indicator for innovation. In the final column the effect of SOA on communication is measured.

Table 3 Measurement and intend

	Empowerment				Performance				SOA
	Impact	Competence	Choice	Meaningfulness	Behaviour	Productivity	Predictability	Innovation	Communication
6.1 Instant messenger analysis		V	V						
6.2 Instant messenger excerpts	V			V				V	
6.3 Release comparison							V		
6.4 Ticket system backlog								V	V
6.5 One case magnification					V	V			

8.1. INSTANT MESSENGER STATISTICAL ANALYSIS

A statistical analysis on instant messenger can be used to reveal cognitions of competency and having a choice at the employee. Behaviours that can be traced to these cognitions are frequent in the IM conversations, therefore it was decided that a statistical analysis is the best way to find a change in behaviour.

Two types of behaviour expressed in discussions could be interpreted as signs of feeling incompetent.

First, a discussion that follows from a question from the offshore employee about “how” something should be implemented. A higher frequency of these questions indicates an incompetent feeling compared to the manager. You would not ask this question if you don’t think the other person has this information. We expect to find more how related questions at the start of the research.

Second, a discussion that follows from the question whether the coach wants to “check” if a certain assignment was done correctly. A higher frequency of these questions makes an

incompetent feeling and/or the feeling of having a choice in executing a task less likely. A competent feeling person is less likely to ask these questions. It could be the person does feel competent; however then it would be more likely that he or she thinks there is no choice in how the task should be executed.

A discussion that follows from the proposal of a solution that concerns the implementation of new functionality indicates the feeling of having a choice in executing a task. If one does not feel to have a choice in executing a task, that person would likely try to understand what the customer wants instead of what the customer needs. We expect to find a higher incidence of these questions towards the end of the research period.

- 1a.** The conversations at the end of the research period will contain less questions on how a task should be executed
- 1b.** The conversations at the end of the research period will contain less questions on checking the correctness of an executed task
- 1c.** The conversations at the end of the research period will contain more solutions to problems concerning the business processes

RESULT

Table 3 shows the nature of 10 conversations in the first 1000 lines of code. Every 100th line we detected the nature of the conversation. The conversations that came out of the random check in the first period only contained questions that relate to how requirements should be implemented. At the end of the research period the employee mostly provided solutions that relate to the business goals directly.

Table 3 Nature of conversations with employee

	10 conversations at start	10 conversations at end
Provides a solution	0	5
How	6	1
Check	0	1
Other	4	3

8.2. INSTANT MESSENGER EXCERPTS

To discover the two remaining dimensions of empowerment, 7000 lines of instant messenger history was deliberately searched for signs of perceived impact and meaningfulness in their

behaviour, over the full extend of the research period. These dimensions could not be discovered using the statistical approach, because the frequency of expressed behaviour related to impact and meaningfulness in the IM history was too low for reliably using random excerpts.

Impact: When you perceive that the work you are doing has a significant impact on the resulting product, you will find yourself feeling responsible for the correct execution of that task. Task that have a high impact on the realization of a shared vision are a liability when they are not executed well. We expect to find a better perception of the products quality.

Meaningfulness: The feeling of meaningfulness in a task assessment will result in the expression of enthusiasm about the work. The search will likely reveal expressed feeling of respect towards the coach, the project and the methodology.

RESULT

In excerpt J2.6 in appendix I, the employee is afraid that another type of transactions in the system will be confusing for the helpdesk. It is the first big client, and therefore very important. He expresses how important he thinks it is to be extra careful.

This excerpt shows that he feels that the software written for the helpdesk is an important part of the overall process. This makes it likely that he feels his work also is an important part of the success of PayDutch. However, this evidence is only indirectly related to the work of the employee. It could be that he does not feel that his work was a significant part of this software.

J2.1 contains a proposal to extend functionality of the Helpdesk. The PayDutch escrow transaction can come into a locked state, which indicates the need for helpdesk intervention. The employee identified the problem that after helping the customers, the helpdesk should be able to unlock the transaction.

This makes it more likely that the employee did understand that his work was important to the business process. If he did not perceive his work as important to the helpdesk process, it would be unlikely that he went beyond the technical problem domain to solve problems.

Another reason for becoming careful could be that you perceive that there will be severe private consequences unrelated to the business goals to incorrectly executing a task.

However, in this case there is no reason to assume that this is the case. The employees behaviour would be suppressed; he would fear the company rules, not the company's failure.

8.3. RELEASE COMPARISON

To find evidence for an increased performance two releases were compared. An important difference between the two releases was the maturity and stability of the methodology, which should positively influence the performance.

Predictability: We expect a higher predictability due to the employee's increased ability to comprehend the problem. The shared vision enables the employee to intuitively distinct details from more important functionality; and thus the employee will be able to make a more accurate planning. SOA prototyping helps in stabilizing the requirements early, which stabilizes the development phase.

Cost: we expect lower implementation cost because empowerment decreases span of control and increases productivity. The client's cost of labour is eight times higher than the vendor's. Less span of control means less cost on onsite labour.

RESULT

The releases under comparison had many similarities besides the usage of RADOS at the start of the project, which makes RADOS a likely contributor to measured improvement. The developers had similar project experience; there was only experience with single developer project without a clearly defined software development process. Both cases started with teams that had no prior experience with Paydutch. Both teams were new to the use of SOA, SVN and the use of online issue tracking in a software development project.

The software process was stabilized in the Toolbox 55 Release. There was a better line between design, prototyping and implementation. Before they started, the whole team met to discuss the projects vision and had a proper training in using RADOS (J1.11). They discussed the techniques in PayDutch together with the employee. The design was validated by the coach once, after correcting it, the prototype was build and presented to the customer. And then the software was implemented.

Table 4 compares the data of the two projects. The offshore effort was measured in weeks per use case. The onsite effort was measured in tickets per use case because we had no data on the amount of time that was dedicated to the project.

Table 4 Additional project characteristics

	Escrow	Toolbox 55
Use cases	28	20
Method at start	None	RADOS
nDevelopers	3	1
Offshore effort		
Design week/use case	1.10	0.20
Development week/use case	2.04	0.25
Team	Asia SD	Teamwish
Onsite effort		
Tickets/uc	3.0	0.4
Vision and high level req.	-	1 Page

The developer had to learn the RADOS methodology from the start, which means that the results taken from this experiment include the time to learn about the company, project and about RADOS.

With this in mind, we can identify other factors that might be of influence on the measured result besides RADOS. We identified the following: Better capabilities; received help; the existence of a reference product.

8.4. TICKET SYSTEM BACKLOG

To find more evidence for an increase in performance the backlog of the collaboration system was scrutinized, containing 411 tickets accumulated throughout the research period of nine months.

Assigning in higher abstractive levels is a sign of improved problem understanding. Abstraction means leaving out details; it requires an expert to understand details from the

more important issues. An employee that better understands what is important to the customer will thus be able to execute tasks that do not include technical details; direct communication becomes possible. Therefore the level of abstraction that the tickets are communicating is likely to increase as the methodology matures.

In May the team changed to a more standardized approach to implementing SOA, which improved communication. It is expected that this change can be measured using ticket abstraction and the amount of information necessary to communicate in the tickets.

This leads to the following hypotheses:

- 4a.** Towards the end of the research period less technical details will be included in tickets assigned to the offshore developers.
- 4b.** Towards the end of the research period less assignment text is necessary per ticket.
- 4c.** The amount of defects found offshore is likely to increase towards the end of the research period.
- 4d.** After May there will be decreased amount of information necessary in the assignments and the information will be of higher abstraction.

Table 5 shows how abstraction was attributed to each ticket during research. Examples of these tickets can be found in the appendix.

Table 5 Tickets containing new functionality

Abstraction	Characteristics
1	Problem explained on code level. Ticket contained implementation code, interface definitions.
2	Problem explained on functional level. Specifications were communicated in the ticket.
3	Problem explained as a vision. A design was requested, like screen proposals, use case diagrams and state charts.

RESULT

Figure 7 shows that the total amount of characters written by the client decreased over time. Also the level of abstraction the tickets were written increased.

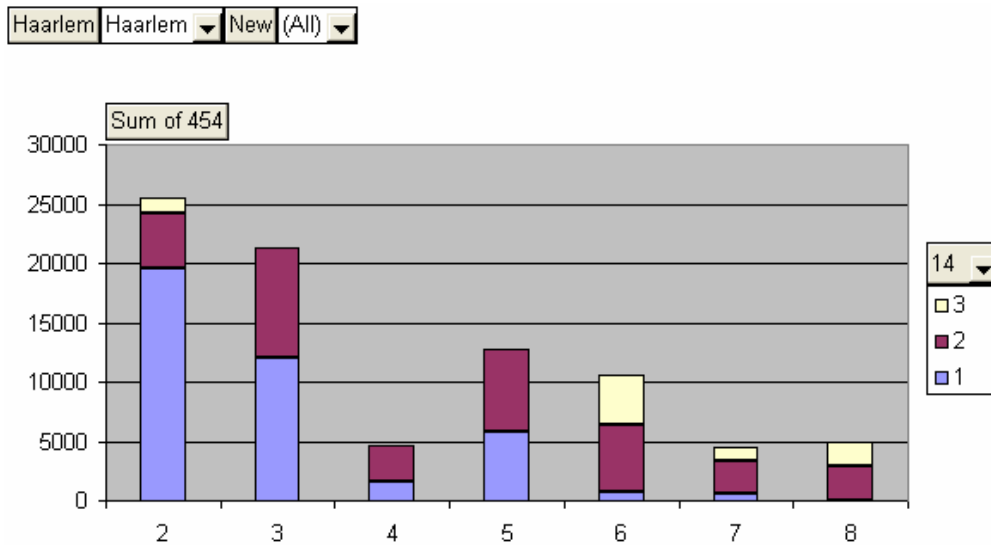


Figure 7 Characters written by the client categorized by level of abstraction

In the fourth month there is a drop in the amount of tickets that were created, the main cause of this was a holiday of two weeks for the chinese new year during this period.

In the fifth month the amount of assignment information is also lower, the reason for this is that the client visited the offshore vendor in china, and a week was spent on developing new methods to implement SOA.

In the sixth month and after that significantly less tickets were and less technical details (more abstraction) in those tickets were needed to assign work to the offshore vendor.

8.5. ONE CASE MAGNIFICATION

This research was conducted on one case where the employee worked on a task that had an empowering effect on him according to his expressed behaviour through IM. This relates the effect of a positive task assessment directly to productivity.

Also behaviour that is linked to empowerment [Thomas & Velthouse, 1990] is expected to be found in the case backlog. Below is a list of all behavioural aspects that result from empowerment and how they should relate to the research data.

Activity: the increased amount of activity should result in an increased amount of code checked in. Also this should not compromise other activity significantly because that could mean that the employee just shifted his attention.

Concentration: The increased amount of concentration should result in the employee's ability to work longer. As the working day progresses, it becomes increasingly difficult to focus. Increased concentration should result in a longer focus. Therefore, code check-ins can be found later on the day.

Initiative: The increased amount of initiative can be found in the employee's ability to foresee potential problems and act upon them. This can possibly result in autonomous problems solving behaviour that can be found in the IM backlog.

Resiliency: This means the employee can endure setbacks better. Possibly setbacks can be found in the conversation between the coach and the employee. When possible, the setbacks should be overcome without the help of the coach. Also these setbacks, together with long working hours make resiliency more likely.

Flexibility: Besides coding, the employee is likely to execute other related work that is needed to successfully execute the task.

Also we expect to find evidence that there was no pressure from the management to put so much effort in this task. If this were the case, the motivation would not be intrinsic as is the case in empowerment.

Furthermore, there would be no significant decrease in other activities, because then the increased amount of effort measured could be the result of a shift of attention.

RESULT

The conversation with the manager before this task was executed is captured in excerpt [D2.1]. The excerpt shows that the employee has a predefined implementation plan. He has no questions related to the implementation of iDeal.

In excerpt D2.2 the employee request our merchant ID, which he needs for attaching iDeal.

In excerpt d2.3 the employee is talking about implementing a payment service using a third party. He is explaining the coach what needs to be done to make this implementation successful. The employee had solved all problems by himself. He requested an account from

a third party; he found out how to generate a digital certificate that was needed to make contact; he made himself a technical contact for this implementation.

In fig. 10 is shown how the number of code check-ins per week increased dramatically after he adopted this task. This is 5 times higher then the mean amount of check-ins in the past.

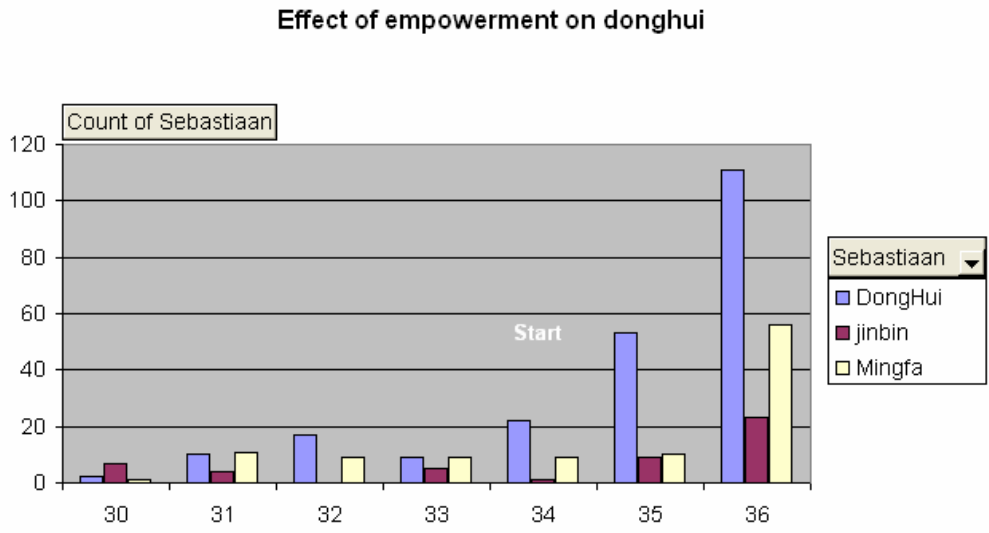


Figure 10 Checkings increase when task assessment is positive

In fig 11 is shown that the number of tickets the employee created during the period his work was researched.

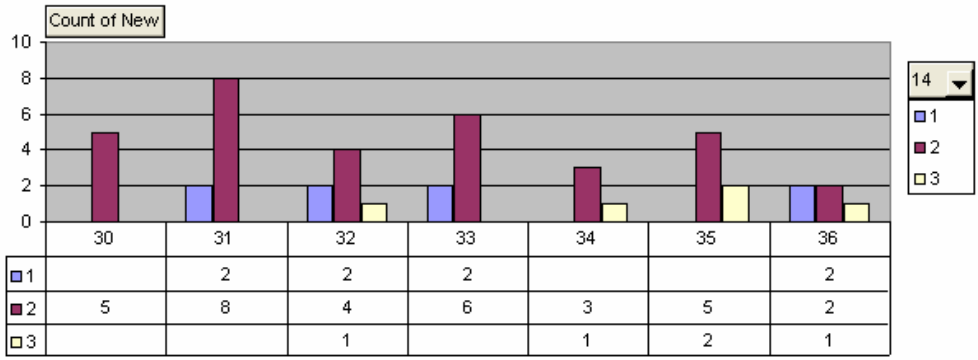


Figure 11 Other activities remained relatively stable

Figure 12 shows that during the increased amount of code check-ins, the employee was working longer then the working day and during that period he worked harder.

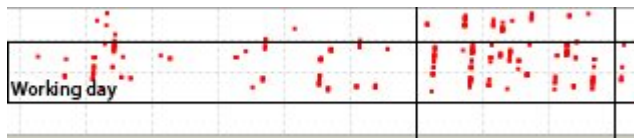


Figure 12 code check in activity during ideal implementation

The increased amount of code check-ins was likely due to increased activity. Fig. 13 showed that during the period the employee was working on the task; the check-ins were more frequent and continued longer.

The fig 11 showed the effort put into creating tickets were more or less the same as before he took the assignment. This makes it more likely that the employee became more active and was not just shifting attention from other tasks.

We can see in Fig. 12 that he checked in more, and he checked in later. That the check-ins are evenly scattered throughout the whole period proves that this increase is not the result of a onetime excessive amount of check-ins.

The later check-ins are likely caused by increased concentration. Working these long hours on the same task is unlikely when concentration is low.

It could be that he changed between different tasks. However, this is unlikely because it costs time to find focus again and that would compromise the total number of check-ins.

The excerpts show that the employee expressed initiative. In D2.1 the employee had a pre-defined plan in place. All steps were sequentially followed according o RADOS procedure. The coach did not have to interfere.

The expected evidence on an increase in resiliency was found. Many setbacks were solved autonomously. It included: the merchant ID could not be used because it was attached to a different account; the certificate did not work on the staging server; and synchronizing branches of the central code base provided problems. These setbacks did not prevent the employee from working until the middle of the night almost every day and were all solved autonomously.

Furthermore, the employee showed to be very flexible. He took on many different subtasks of which he had no experience and that required him to build skills beyond developing code. These were: Code branching and merging in SVN; installing a certificate; creating an iDeal account.

9. MEASUREMENT RESULT

The measurement result will be discussed in the following subparagraphs. First we discuss in what amount we found evidence for the existence of empowerment. Second, we argue that there was an increase in performance and third whether this increase in performance can be related to the implementation of RADOS.

9.1. DOES RADOS RESULT IN EMPOWERMENT?

Empowerment is defined in 4 dimensions, and therefore this research questions should also be answered in those four dimensions.

“The Instant messenger random check” analyzed 20 conversations of the employee over a prolonged period of time and showed that conversations at the end of the research period provided more solutions, whereas at the beginning of the research period the conversations contained more questions on how something should be implemented and whether the implementation was done correctly.

“The Instant messenger random check” showed 5 occurrences in which the employee provided a solution to a problem regarding higher level business processes.

The instant messenger excerpts showed us that it is likely that at least three employees were proud on the work they were doing. Evidence was provided that the employee had a general feeling of doing meaningful work towards the end of the research period, and that this is likely based on his perception of the vision and values of PayDutch. Furthermore, he explained how other team members were proud on working with Paydutch, which could of course be an occasional expression, but still it makes their feeling of doing meaningful work more likely.

The instant messenger excerpts showed us that the employee’s care went beyond his role as a developer; he was actively aware of the fact that care should be taken to prevent problems. This cautious behaviour is likely to result from the feeling of doing important work.

The one case magnification showed another employee’s expressed behaviour that resulted from a positive task assessment was according to the expected behaviour in the model of

Thomas & Velthouse [Thomas & Velthouse, 1990]. This supports the theory by making it likely that at least two of the employee's were empowered.

RESULT

According to this research, at least one employee felt and acted empowered in all 4 dimensions of the definition. The result of the "The Instant messenger random check" provided evidence that an employee felt competent and that he had a choice in how to do his work. The result of "The instant messenger's excerpts" showed that he had a general feeling of doing meaningful work which is important.

That one employee felt and acted empowered, makes it likely that the larger part of the team felt empowered as well. Focusing only on empowering one employee is not likely to be effective using only IM, while the employee's direct environment is continuously counterbalancing the influence. In the one case magnification this is supported by the fact that the employee showed all behavioural aspects that result from empowerment according to literature [Thomas & Velthouse, 1990].

Furthermore, the empowered employee referred to the other employees as being proud to work for PayDutch, which indicates that they feel they are doing meaningful work. This could be based on occasional behaviour, but the supporting evidence so far makes this unlikely.

9.2. SERVICE ORIENTED ARCHITECTURE AND COMMUNICATION

In 7.4 the ticket system backlog was measured that after May communication in assignment tickets became less and more abstract. After a visit of the client to China in May, the team had learned a lot about SOA and that was the moment to change the approach to building SOA services. This is a likely cause of the improvement measured in assignment communication.

Other factors can be of influence to the improved communication, like the improvement of product knowledge during development or better understanding of the companies goals, but the rapid improvement after May made it likely that the change of approach was a large contributor to this improvement.

9.3. RADOS PERFORMANCE

The evidence was not easy to compare to a similar case. RADOS was created incrementally, and when it was created, the assignment that followed was related to the overarching

architecture. The latter had a better chance of finding a similar case. But the researcher decided to reduce the scope at this point due to the time it would cost to exactly measure the invested effort.

The comparison of the release at the beginning and that at the end of the research period showed that the amount of hours worked on a use case offshore and the amount of tickets created per use case was both lowered about 80%.

The ticket system backlog showed an increased level of abstraction in new assignments from the client was accompanied by a coarse 80% decrease in assignment size between the eight and the second month. It is likely that time spent on assignments onsite decreased with a similar percentage, which is a strong indicator for cost reduction.

The ticket system backlog showed that the developers actively participated in the testing phase and performed most of the structured tests. However, during the acceptance phase a large number of defects were still found offshore and onsite.

The one case magnification showed that the employee's expressed behaviour that resulted from a positive task assessment was according to the model of Thomas & Velthouse [Thomas & Velthouse, 1990], it also provided evidence for an increase in productivity.

RESULT

The research provided evidence that the implementation of RADOS had a positive effect on some performance related variables. These are discussed below.

Predictability: The comparison of the release at the beginning and that at the end of the research period showed that predictability was increased according to the first release. A new team used RADOS and delivered the product only two days late. There were some factors that could influence this outcome that are difficult to measure from project backlog, but it can be regarded as at least a strong indication that predictability improved because of RADOS.

Cost: the research data made it likely that cost reductions after implementing RADOS were significant. The cost per use case was lowered onsite and offshore. Of course there are external factors of influence of this data, but at least this provides a strong signal that productivity increases and effort in work assignment by the client is strongly diminished. This

is also supported by the “One case magnification” where a spectacular productivity increase could be related to empowered behaviour.

Innovation: The increased number of solutions related to the high level business processes came from the vendor is an indicator for innovation. This also is known in research as a result of empowerment [Mao, 2007].

10. CONCLUSION

Developers in low wage countries are very motivated to do their work right. In currently used methodologies we are not able to exploit this. Therefore many companies are not able to gain a significant return of investment and therefore often stop offshore development after their first attempt.

Instead of addressing the problems and risks onsite, RADOS uses the motivation of the developers by enabling them to see and mitigate risks themselves. The work done offshore becomes more inspiring and rewarding. The developers are motivated to build the right software instead of building the software right.

The model of empowerment used by Thomas and Velthouse makes clear that the employees perception of a task is largely based on environmental events in the employees workplace. In RADOS part of this environment is the teams coach, who will help the employees in their personal growth and work on building a shared vision amongst all team members. Sharing a vision with a competent feeling employee and viewing a challenging task as part of that shared vision, results in productive behaviour and innovation. Building a shared vision takes time, but ultimately pays off because it enables the employee to make more and better decisions based on higher level goals. Another important part of this environment is the software architecture. RADOS keeps the software comprehensible by integrating standardized patterns of service oriented architecture in the working process.

10.1. RESULTS

The result of this research was RADOS, a validated offshore software development methodology that enables empowerment in offshore development. The cognitive elements of empowerment were found in one case, the expected behaviour was found in another case. The overall tendency in the instant messenger history was very positive towards the tasks and the coach. It is very likely that the whole team was generally empowered.

SOA contributes to communication in offshore development; the size of assignments decreased significantly after the introduction new methods to implement SOA services. It was experienced as building several smaller applications that communicated well together instead of one large application. According to expectations SOA development improved scoping and so enabled discussion about the application from a higher perspective.

This resulted in an increase in productivity, predictability and autonomy. Which was measured in two separate cases. The experience of the developers at the start of the project was very low. The offshore vendor only did single developer projects; another project done by the offshore vendor failed; the developers told the coach that they learned a lot. It becomes likely that RADOS caused these results.

An unexpected result was the importance of a cultural fit between the client and the offshore vendor. The initial vision did not include this, but later on it seemed as one of the most important collaborating aspects. Finding a powerful fit between cultures was found to be vital to the project's success. In this research trust was articulated as an important company value, which is an incredible weak factor in Chinese culture, but of the outmost importance to software design and development. Making trust an important focus point and celebrating the Chinese ability to collaborate in contrast with the client's native culture helped in establishing a shared culture, which became a strong carrier for effective communication.

10.2. AGILITY IN OFFSHORE DEVELOPMENT

RADOS can be categorized as an agile approach. Although many commonly used agile practices are not used in RADOS, it does fulfil the basic requirement of being responsive to changing requirements.

One important practice in agile approaches which is not contained in RADOS is real customer involvement. Including the customer throughout the whole development process is important to agileness. It enables the developers to capture requirements based on requirements changes that emerge throughout the development process. When the customer changes its mind, the developers are able to flexibly respond to that.

To overcome problems that lie tacit within the problem area of the customer, RADOS tempts to communicate the needs of the customer on the highest possible level of abstraction. The information communicated becomes more visionary and problem related. The developers will develop a problem framework that relate to that in the mind of the customer which will provide them with the ability to make decisions that closely resemble to the decisions the consumer would make. The paper prototype will reduce risk when the developer misperceives the customers vision.

The remaining problems were solved by coaching the offshore developers to directly communicate with stakeholder through e-mail and instant messenger. Towards the end of the research process this approach became very successful. Developers were directly communicating with important stakeholders in large organizations like the head of ICT from TNT Benelux, the largest shipment company in the Netherlands; and with the support desk of Equens, a major payment cooperation in Europe.

10.3. FUTURE WORK

The results of this research were inspiring; RADOS increased performance considerably while being in an early stage of development. RADOS only scratches the surface of the possibilities that come with this approach; only basic knowledge of SOA, empowerment and the Chinese culture was used. Specializing and improving on these knowledge areas will further increase performance.

Designing a course that help the offshore employees in communicating with stakeholders can significantly improve the effectiveness of communication. The developers became increasingly capable in directly contacting stakeholders. For example there was much direct contact between one of the developers and partners that needed software integration. This contact became more effective over time, and eventually the whole development team learned to communicate with foreign stakeholders of the software product.

Also there are still numerous ways open to improve comprehensibility by using state of the art tooling used in SOA development. Since the past two years, a lot of tools are published that make SOA development better to comprehend. The adoption of windows communication foundation made a big difference in comprehensibility because all the best practice patterns were automated, which provided clarity in the separation of concerns. Currently RADOS is integrating windows workflow foundation, which provides clear graphical insight in workflow dynamics in businesses. This will make discussion about the code more open and therefore improves common understanding between onsite and offshore team members.

The findings in this research resulted in the start of a new company, named 'Second Company'. The idea behind RADOS still is very basic, and needs maturation before it can be handed over to the greater public. A commercial company will provide the environment and funding that is necessary to do so. RADOS will be subjected to working practice in real world scenarios. As more employees are going to work in this new way, the methodology will become more formal until it is ready to be handed over to the greater public.

REFERENCES

- Agarwall - "The Maturation of Offshore Sourcing of Information Technology Work", Springer Berlin Heidelberg, 2006
- Aspray, W. - "Globalization and offshoring of software", ACM, 2006
- Automatiseringsgids, "Nieuwe stijging vacatures leidt tot krapte", Het Financieele Dagblad, 2007
- Beck, K. - "Extreme programming explained" 2nd edition, Addison-Wesley, 2005
- Beynon-Davies, P. - "Rapid application development (RAD): an empirical review", Operational research society Ltd., 1999
- Bhat, J.M. - "Lessons from offshore outsourcing", Infosys technologies, 2006
- Boehm, B. - "Anchoring the software process", IEEE, 1996
- Boehm, Ch. - "What makes IT offshore different", Transcrit, 2003
- Bundy, R.A. - "Create cultural change to support a business transaction", Mercer – Issue 3 human capital, 2007
- Christiansen, H.M. - "Case Studies On Bringing Agility to Offshore Software Development", Agile Journal, 2007
- Clancy - "The CHAOS Report", The Standish Group International, 1994
- Damian, D. - "Awareness meets requirements management: awareness needs in global software development", Dept. of Computer Science University of Victoria, 2003
- Eidson, B. - "SOA and the future of application development", Oracle corporation, 2005
- Fergusson, E. - "Offshore Outsourcing: Current Conditions & Diagnosis", Norfolk, 2004
- Goldenson, D.R. - "Demonstrating the benefits and impact of CMMi", Carnegie Mellon University., 2003
- Landis, K.M. - "Calling a Change in the Outsourcing Market", Deloitte Consulting, 2005
- Levi, K. - "A Goal-driven Approach to enterprise component identification and specification", ACM, 2002
- Mao, J.Y. - "Capabilities building in Chinese software services firms", Val d'Isère France, 2007
- McDougall, P - "IBM To Invest \$6 Billion In India To Increase Offshore IT Services Offerings", Information Week, 2006
- Kroghdahl, P. - "Service oriented agility: an initial analysis for the use of agile methods for SOA development.", IEEE, 2005
- Monsterboard
- Minevich, M.D. - "The Global Outsourcing Report Opportunities, Costs and Risks", CIO Insight, 2005

Natis, Y.V. - "Service oriented architecture scenario", Gartner inc, 2003

Nonaka, I. - "The knowledge creating company", Harvard business review, 2007 (First published 1991)

Rajkumar, T.M. - "GSD, The view from indian suppliers", Taylor & Francis, 2001

Rhongzu K. - "Trust in China: A Cross-Regional Analysis", The William Davidson Institute, 2003

Santos, J. - Building Software Factories - Part 1, what are we building and why?", Microsoft press, 2006

Shamsi, A. - Offshoring: forget cost reduction focus on quality, 2007

Simons, R. - "Control in an age of empowerment", Harvard business review, 2007 (First published 1995)

Spreitzer, G.M. - "Psychological empowerment in the workplace: Dimensions, measurement, and validation" , Academy of management Journal, 1995

Spreitzer, G.M. - "Social structural characteristics of psychological empowerment" , Academy of management Journal, 1996

Thomas, K.W. & Velthouse, B.A. - "Cognitive elements of empowerment", Academy of management review, 1990

Watanabe, K. - "Lessons from toyota's long drive", Harvard business review, 2007

Wood, J. and D. Silver - "Joint Application Development, 2nd ed.", Wiley, 1995.

Woolf, B. - "Streamline SOA using service mocks", IBM, 2005

Zimmerman, O. - "Analysis and Design Techniques for Service-Oriented Development and Integration", IBM, 2005

Appendices

I. EXCERPTS JINBIN

Choice and Competent | J1.1 (4/6)

...

Jinbin: what is job my service do?

..

Sebastiaan: it fills in online webforms automatically

Jinbin: such as the ftp provider file trans

Sebastiaan: by feeding parameters

Jinbin: ok

Jinbin: the parameter is pass in by client?

Sebastiaan: yes

..

Jinbin: I using the function your webconnector provide or make a new?

Sebastiaan: not new

...

Jinbin: okay, then I just package it as a wcf service?

Sebastiaan: yes

..

Jinbin: I provider data contract for parameter?

Sebastiaan: yes

...

Choice | J1.3 (4/12)

...

Jinbin: I am programing to implement the IHelpdeskFacade

Sebastiaan: yes

Jinbin: and I find we are want a user manager component

Jinbin: user that we can validate the login user and manager the user infomation

Jinbin: how do you think about it?

Sebastiaan: I agree

...

Sebastiaan: let's push this requirement a little backwards

...

Understand expectations | J1.4 (4/7)

Sebastiaan: can you also discuss the problems of WCF with Dong Hui? He is responsible for quality, i think he should now, he also has a clear vision of the whole project.

Jinbin: yes

Sebastiaan: maybe setup a little meeting or so

Jinbin: I have told him last Friday

Sebastiaan: you can discuss how to implement this best

Sebastiaan: ah, all right

Jinbin: and he told me the all the vision about the project

Jinbin: he is very zealous

Sebastiaan: this implementation you are now performing is a test, before you start, you and donghui must make a strategy on this

Sebastiaan: yes i know :)

Jinbin: I will get along with him very glad

Choice | J2.1 (5/31)

Jinbin: I think the HelpDesk must have a UnLock Function. This function can change the transaction status from unlock to inspect or retrieve or compensate

Jinbin: how the helpdesk change the transaction status from locked to other status?

.....

Sebastiaan: UnLock! Like a big button somewhere

.....

Jinbin: Yes, you get it!

.....

Meaningful | J2.2 (6/30)

Jinbin: some point of the document is right. just because the enterpriser and merchant are less education in school. They do most work by themselves or their relative. So they only trust the practice.

...

Jinbin: but more and more Chinese which have high education in university, they have another point about trust.

...

Jinbin: they feel that China need build a trust system, just like European now does. Then China's economic can develop better. And trust system is build one by one.

...

Jinbin: current trust system let rich more rich and poor more poor.

Jinbin: poor have little change to success.

...

Jinbin: so, my friends mostly are poor like me. And we trust each other. :)

...

Sebastiaan: Nice, well, i am sure you succeed. And that when you did succeed, you won't become a victim of the luxuries and emptiness of a rich mans life, but you will spread a new culture among your rich friends that enables your environment to become friendly for people in all layers of the Chinese community.

...

Jinbin: it is my dream

Jinbin: change China to be a friendly country.

Choice | J2.5 (9/12)

...

Jinbin: I need he understand well the state flow

Sebastiaan: I think you are right

Sebastiaan: but flow in the requirements document is that right?

Jinbin: yes.

Jinbin: we just description same thing in difference way, make it more clear.

Sebastiaan: ah

Jinbin: your way is good and beautiful. Mine is other way to describe it.

Sebastiaan: ah, and it is short and readable

Jinbin: maybe it is Chinese way.

...

Impact, Meaningful | J2.6 (9/12)

Jinbin: if the tbx55 publish, it will use the same db of escrow. But maybe effect current escrow, and how do you think it?

Sebastiaan: I think it is good that you think about this, because we should take measures

...

Jinbin: and I hope helpdesk can do some test. It will make them more sense about it.

Sebastiaan: yes, donghui is corresponding with jo
Sebastiaan: you are right
Sebastiaan: jo is helpdesk
Jinbin: because it is a new business and the first big client business too.
Jinbin: okay.
Jinbin: that will be very well.
...
Jinbin: care, care, and care again.

Meaningful | J2.7 (9/8)

...
Jinbin: yes. they all are very thank for your email
Jinbin: guoqi and shawn feel very good at it.
...
Jinbin: shawn want to use some method of CMMI for the Asia SD.
Jinbin: but donghui and mingfa think they are fortune in the PayDutch
Jinbin: they all think the PayDutch project manage is better than it.
...
Jinbin: feel proud to work with you and at PayDutch
...
Jinbin: but It is begin from you. Thank you very much
Jinbin: I learn too much from you at a short time.
...

Trust | J2.9 (7/16)

Jinbin: one thing, I should tell because you are my friend.
Sebastiaan: yes?
Jinbin: Guoqi query Jinfu if you depend on us very much.
Sebastiaan: okay, what does he ask then?
Jinbin: he said, if you depend on us very much, then he will have some method to treat to you.
...
Sebastiaan Makes me sad, but we cannot change the world in one day my friend :)
Sebastiaan: we need a week at minimum
Jinbin: but I don't know Jinfu's answer.

Choice | J2.10 (7/6)

Jinbin: for my opinion, we can try to let consumer and merchant do some compensation. If they agree each other, the helpdesk can do nothing. and if they are locked, then helpdesk just read the compensation info, and make a decision.

Jinbin: for my opinion, mostly consumer and merchant will make an agreement. So helpdesk's workload will be less.

Sebastiaan: You can be right about that!!

Sebastiaan: But we need another tool for that, not to compensate.

Sebastiaan: you agree?

Jinbin: yes. Such as negotiate.

.....

Understand expectations | J2.11 (7/8)

...

Jinbin: as a matter of fact.

Jinbin: just because I want he do more job at the project.

Jinbin: so, I let he go faster .

Jinbin: I want left sometime to meeting and study

Jinbin: this project's mainly task is studying.

Sebastiaan: aha

Jinbin: we have not use the RADOS before.

Jinbin: so I want our team to study after do something.

II. EXCERPTS DONGHUI

Autonomy | D2.1 (8/13)

...

Donghui: first, i must read the idea documentation. Second, we will make screen, use case, role description, then generate analysis or solution for this.

Donghui: third

Sebastiaan: okay, I am listening

Donghui: connection with downloadable code, collaboration with Mingfa how to implement it, and keep ideal release in good condition, including readability and good quality

...

Initiative, activity | D2.2 (8/15)

Donghui: hi..

Sebastiaan: hi

Donghui: may i have the merchantID for ideal??

Donghui: now i am testing my application..

Sebastiaan: you what/

Donghui: the merchantID like 008026150

Resiliency, flexibility | D2.3 (8/17)

Donghui: now I can log in

Sebastiaan: (Y)

Donghui: signup process in the requirement of ideal integrated.

Sebastiaan: shall I make you technical contact person?

Donghui: of course.

...

Donghui: now we are in step 4, have you received contract sent by Fortis ideal

Sebastiaan: I don't know, can you not test now?

Donghui: you can see the Fortis ideal signup process

...

Sebastiaan: contract is not necessary I think

Donghui: yes, if we receive the contract, then we will upload the certificate

Sebastiaan: oh

Sebastiaan: ok

Donghui: I know how to generate certificate, private and public key

Donghui: and yesterday I heard from Sharif that he received a contract.

Donghui: we will upload certificate in configuration menu

Sebastiaan: what do you need me to do?

Donghui: you have already done a great help to me...

Donghui: now I will generate certificate

Sebastiaan: :D

Sebastiaan: did not do anything

Sebastiaan: but thank you

Donghui: haha...but at least I could log in.

Donghui: thank you anyway

III.TICKETS

Abstraction 1 in period 1

Consumer interface Opened 8 months ago
Last modified 6 months ago

Reported by:	Timegrip	Assigned to:	Linfu
Priority:	high	Milestone:	
Component:	Website Customer	Version:	
Keywords:		Cc:	
Time spent:		Time remaining:	
Time planned:	4d		

Description (Last modified by anonymous) Reply

Login page

[View image](#)

This page creates an Timegrip.Paydutch.[TransactionTicket](#) that can be used to access the [ConsumerService??](#). This [TransactionTicket](#) should be maintained while the user is logged on to the application. Using the transactionticket the user can load a transaction from the consumerservice. The next control depends on the escrowstate in the transaction (see the paydutch [Architecture](#))

Validation::

Phonenumber starting with 06 or 04
Transaction code, on the [TransactionTicket](#)

Step 1

[View image](#)

1: Escrow.Name {string: 50}
2: Escrow.Code: {string: 30}
3: Escrow.Price: {decimal: 18,2} - No euro sign in the textfield
4: Escrow.Description: {string: 512}
5,6,7,8: Should display the help text as in fig 2.

When the input requirements are not met, fig 3 should be shown. The information needs to be on a layer in front of the screen. The sign under 5, 6, 7, 8 should change into an yellow alert sign (not included).

9: A separate control that displays the different steps a consumer needs to walk through.
10: Goes to the login screen [1 Login]
11: If all fields are filled, when pressed this button, the user will be redirected to [3 Registration info]

Step 2

[View image](#)

Abstraction 2 in period 1

Test: login page for the maintenance interface		Opened 8 months ago Last modified 6 months ago	
Reported by:	RobertJan	Assigned to:	DongHui
Priority:	medium	Milestone:	
Component:	Website Maintenance	Version:	Review
Keywords:	Maintenance interface login	Cc:	Zuhua, RobertJan
Time spent:	1	Time remaining:	
Time planned:	1		

Description (Last modified by RobertJan) Reply

Before: Fill in 'Time planned' After: Fill in 'Time spent'

This is a quick fix just to get the maintenance interface working. Don't worry too much about the layout. There will be another ticket for the layout later.

Use the same master page and page style you use for the Gatekeeper page etc. See ticket [#21](#) for this.

The Page

- The page should be multi-lingual with English as the first (and for now only available) language. So implement it in English but make it easy to add other languages.
- The page has 4 input controls User, Password, Role and Scenario
- user and password are textboxes as usual
- Role and Scenario are dropdowns
- Scenario has only the option 'Mock' for now
- Role has the options 'Gatekeeper', Helpdesk and 'Web manager'
- There is a 'Login' button for submitting the page.

Action on submit

- A ticket will be created and tested against the relevant webservice. A call 'bool `ValidateTicket?()`' will be added to the web services in Maintenance for this purpose.
- If the call returns false the page should clear the password textbox and display the error message 'Authentication failed'. The user can then try again.
- If the call returns true the user should be shown the startpage belonging to his role (gatekeeper/default.aspx, helpdesk/default.aspx or webmanager/default.aspx).
- Make stubs if these pages don't yet exist.
- The ticket should be kept in the session for further calls to the web services. Test this by modifying the Gatekeeper page to use the same ticket.

Abstraction 3 in period 2

Improve compensation proposal tool

Opened **3 months** ago
Last modified **3 months** ago

Reported by:	sebastiaan	Assigned to:	jinfu
Type:	Change	Priority:	High
Milestone:	The Escrow Release	Component:	Consumer Interface
Version:	Implementation	Keywords:	
Cc:			

Description (last modified by sebastiaan) ([diff](#)) Reply

Before i fire away, i like to give the opportunity to find at least 3 changes that would improve the compensation proposal tools functionality and quality. A big tip would be that you try to stand in the shoes of the customer. What do you think a customer wants to improve on this tool.

So::

1. Improve 3 things
2. From a customer perspective
3. And try to put the implementation time when choosing the improvements.
4. Put your improvements in this ticket

Improvements::

1. Display if the merchant is online, if the merchant is on line, send instant message, else send email to the merchant
2. Clear the sending text area after sending message for next message.
3. Display the last proposed percentage in highlight
4. Create a send button also for merchant instead of decline, it will send the message, nothing else.
5. Enter = send is also a nice feature.

Abstraction 3 in period 2

TIY The prototype documents for TicketYourself; Finished: 08/21/2007 Opened **2 months** ago
Last modified **3 weeks** ago

Reported by:	xiangbin	Assigned to:	Xiangbin
Type:	New	Priority:	Medium
Milestone:	The Ticket Yourself Release	Component:	Teamwish
Version:	Design	Keywords:	
Cc:	Xiangbin		

Description (last modified by sebastiaan) ([diff](#)) [Reply](#)

Task Description, Role Description, Use case Diagram, Screenshots

A presentation will be created by Xiangbin.
Expected date finished: 08/21/2007

Remark::

1. I like the prioritization. Good

Attachments

- [TicketYourself Screenshot.rar](#) (0.9 MB) - added by *xiangbin* **2 months** ago.
- [TicketYourself Role Description.doc](#) (28.5 kB) - added by *xiangbin* **2 months** ago.
- [TicketYourself Task Description.doc](#) (165.5 kB) - added by *xiangbin* **2 months** ago.
- [TicketYourself.vsd](#) (155.5 kB) - added by *xiangbin* **2 months** ago.

[Attach File](#)

Abstraction 2 period 2

Ticket #149 (closed defect: fixed)

validation for zipcode,house number and country in delivery interface. Opened **3 months** ago
Last modified **1 month** ago

Reported by:	donghui	Assigned to:	donghui
Type:	Defect	Priority:	Medium
Milestone:	The Delivery Release	Component:	Delivery Interface
Version:	Implementation	Keywords:	
Cc:	donghui		

Description [Reply](#)

description:

it is necessary to validate for zipcode,house number and country whether they are stored in a database, so making sure zipcode is valid.

Abstraction 1 in period 2

Ticket #109 (closed new: fixed)

mail template in webmanager/consumer/ merchant.		Opened 3 months ago Last modified 3 months ago	
Reported by:	donghui	Assigned to:	jinfu
Type:	New	Priority:	High
Milestone:	The Escrow Release	Component:	Expiration
Version:	Implementation	Keywords:	webmanger
Cc:	donghui		

Description Reply

problem1:

1. in mail content Template, find it doesn't include `Consumer.PhoneNumber?` and `Merchant.PhoneNumber?`.

`mailContent.Replace("{%Consumer.PhoneNumber?%}", consumerMobile); mailContent.Replace("{%Merchant.PhoneNumber?%}", merchantMobile);` consumerMobile and merchantMobile are not evaluated.