

Pontifícia Universidade Católica de São Paulo
PUC-SP

Lucimara Oliveira Corrêa Soares

A importância da comunicação no desenvolvimento de
software em pequenas e médias empresas

MESTRADO EM TECNOLOGIAS DA INTELIGÊNCIA E DESIGN DIGITAL

São Paulo - SP
2017

Pontifícia Universidade Católica de São Paulo
PUC-SP

Lucimara Oliveira Corrêa Soares

A importância da comunicação no desenvolvimento de software em pequenas e médias empresas

MESTRADO EM TECNOLOGIAS DA INTELIGÊNCIA E DESIGN DIGITAL

Dissertação apresentada à Banca Examinadora como exigência parcial entre a obtenção do título de MESTRE em Tecnologias da Inteligência e Design Digital pela Pontifícia Universidade Católica de São Paulo, sob orientação do Prof. Doutor Sergio R. Basbaum.

São Paulo - SP

2017

Banca Examinadora

À minha família, a razão de tudo.

AGRADECIMENTOS

Agradeço a PUC que é parte da minha história desde a adolescência e a todos os professores da graduação que tiveram papel fundamental na minha formação profissional.

Agradeço ao meu amado marido, que me incentiva e me apoia em todas as muitas jornadas que eu assumo na vida. Aos meus filhos maravilhosos que me ensinam muito sobre a vida desde o dia em que nasceram. Ser mãe me fez querer ser uma pessoa melhor, ainda que seja só para dar o exemplo correto. E me ensinou que percorrer o caminho correto, ainda que seja mais trabalhoso, sempre vale a pena.

Eu gostaria de registrar aqui um agradecimento especial a Edna, sempre prestativa e disposta a ajudar a todos.

E também ao professor Ítalo Santiago Vega, que me orientou na graduação e teve papel fundamental na minha banca de qualificação.

E como não poderia deixar de ser, agradeço aos professores do TIDD, especialmente meu orientador Professor Doutor Sérgio R. Basbaum, que me ajudou a acreditar que eu estava no caminho correto quando isso acontecia e me ajudou a corrigir o percurso quando era necessário. Tenho a certeza de que sem ele este trabalho não seria finalizado no prazo.

“Quando tenho a impressão de me ocupar apenas com palavras, é que a expressão falhou; Ao contrário, se ela é bem sucedida, parece-me que penso ali, em voz alta, naquelas palavras que eu não disse”.

Maurice Merleau-Ponty

RESUMO

Esta dissertação se propõe a estudar o papel da comunicação entre as pessoas de uma equipe envolvida no desenvolvimento de software, tratando de profissionais que trabalham em pequenas e médias empresas. Demonstra o cenário atual deste segmento, com apresentação de índices de fracasso na conclusão de projetos fornecidos pelo Chaos Report e, apresenta conceitos da Teoria das Inteligências Múltiplas de Howard Gardner, e conceitos da Transmissão de Conhecimento para demonstrar o perfil dos profissionais. Além disso, mostra um estudo de caso exemplificando um tipo de falha que pode ocorrer, em função do problema da comunicação.

Em seguida, numa busca por entender como se dá a comunicação, apresenta conceitos de percepção e diálogo através das visões dos filósofos Maurice Merleau-Ponty e Martin Buber.

Finalizando, mostra duas técnicas de modelagem, UML e BPMN que se propõem a resolver a questão da falha de comunicação nos projetos de desenvolvimento de software.

Palavras-chave: Comunicação, modelagem de processos, projetos de software, inteligências múltiplas, gestão de conhecimento, UML, BPMN

ABSTRACT

The main purpose of this dissertation is to study the role of communication among professionals involved in software development within small and medium enterprises.

It demonstrates the current scenario of this segment presenting data from projects that were not successfully accomplished providing numbers from Chaos Report. The concepts of Howard Gardner's Theory of Multiple Intelligences and the concepts of Knowledge Transmission were applied to map the profile of the professionals. In addition to that, it presents a case study exemplifying a type of issues that may occur due to communication problems.

Therefore, in a quest to understand how communication takes place, this paper includes concepts of perception and dialogue through the views of philosophers Maurice Merleau-Ponty and Martin Buber.

Finally, it shows two modeling techniques, UML and BPMN that propose solutions to solve problems concerning communication failures in software development projects.

Keywords: Communication, process modeling, software projects, multiple intelligences, knowledge management, UML, BPMN

LISTA DE FIGURAS

FIGURA 1 – SEGMENTAÇÃO DO MERCADO BRASILEIRO DE SERVIÇOS.	20
FIGURA 2 – SETORES DE ATUAÇÃO DAS EMPRESAS EM PERCENTUAIS.....	21
FIGURA 3 - RESULTADOS DOS PROJETOS.....	27
FIGURA 4 - RESULTADOS DOS PROJETOS DE ACORDO COM O TAMANHO DO PROJETO.	28
FIGURA 5 - RELACIONAMENTO EQUIPE-CLIENTE	30
FIGURA 6 - RELACIONAMENTO INTERNO DA EQUIPE	31
FIGURA 7 - PROCESSO SECI. FONTE: ADAPTADA DE NONAKA E TAKEUCHI, 1995.....	46
FIGURA 8 - UMA PEQUENA PARTE DO METAMODELO UML. FONTE: FOWLER (2005, p. 32).....	64
FIGURA 9 - CLASSIFICAÇÃO DOS TIPOS DE DIAGRAMA DA UML. FONTE: FOWLER (2005, p. 148).....	65
FIGURA 10 - DIAGRAMA DE ATIVIDADE SIMPLES. FONTE: FOWLER (2005, p. 119).....	67
FIGURA 11 - EXEMPLO DE TEXTO DE CASO DE USO. FONTE: FOWLER (2005, p. 105).....	68
FIGURA 12 - DIAGRAMA DE CASOS DE USO. FONTE: FOWLER (2005, p.107)	69
FIGURA 13 - UM DIAGRAMA SIMPLES DE MÁQUINA DE ESTADOS. FONTE: FOWLER (2005, p. 111)	70
FIGURA 14 – UM DIAGRAMA DE SEQUÊNCIA PARA CONTROLE CENTRALIZADO. FONTE: FOWLER (2005, p. 68).....	71
FIGURA 15 – UM DIAGRAMA DE COMUNICAÇÃO COM NUMERAÇÃO DECIMAL ANINHADA. FONTE: FOWLER (2005, p. 130)	72
FIGURA 16 – DIAGRAMA DE RESUMO DE INTERAÇÃO. FONTE: FOWLER (2005, p. 140)	73
FIGURA 17 - BPMN BASICS. FONTE: ADAPTADO DE OMG(2011)	77
FIGURA 18 – ELEMENTOS REPRESENTATIVOS DE EVENTOS. FONTE: ADAPTADO DE OMG(2011)	78
FIGURA 19 – ELEMENTOS REPRESENTATIVOS DE ATIVIDADES. FONTE: ADAPTADO DE OMG(2011).....	78
FIGURA 20 – ELEMENTOS REPRESENTATIVOS DE GATEWAY. FONTE: ADAPTADO DE OMG(2011).....	79
FIGURA 21 – ELEMENTOS REPRESENTATIVOS DE FLUXOS. FONTE: ADAPTADO DE OMG(2011)	80
FIGURA 22 – ELEMENTOS REPRESENTATIVOS DE DADOS. FONTE: ADAPTADO DE OMG(2011).....	81
FIGURA 23 – ELEMENTOS REPRESENTATIVOS DE ARTEFATOS. FONTE: ADAPTADO DE OMG(2011)	82
FIGURA 24 – ELEMENTOS REPRESENTATIVOS DE SWIMLANE BASICS (COLABORAÇÃO). FONTE: ADAPTADO DE OMG(2011)	83

Sumário

PRÓLOGO	11
INTRODUÇÃO	13
ESTADO DA ARTE	13
JUSTIFICATIVAS	16
OBJETIVOS E METODOLOGIA	16
CAPÍTULOS	17
CENÁRIO ATUAL DO PROFISSIONAL DE DESENVOLVIMENTO DE SOFTWARE	19
PEQUENAS E MÉDIAS EMPRESAS	21
OTIMIZAÇÃO DE RECURSOS	23
CHAOS REPORT	27
NÍVEIS DE INTERAÇÃO	29
<i>Relacionamento entre equipe de desenvolvimento e cliente</i>	<i>30</i>
<i>Relacionamento entre as pessoas da equipe de desenvolvimento do software</i>	<i>31</i>
ESTUDO DE CASO	35
DIFICULDADES EM TORNO DO PROBLEMA	38
COMPREENSÃO DE UM PROBLEMA	40
TRANSMISSÃO DO CONHECIMENTO	43
COMUNICAÇÃO	49
MODELAGEM DE PROCESSOS	56
UML	61
<i>Documentação através de UML</i>	<i>63</i>
<i>Diagramas</i>	<i>63</i>
<i>Diagramas de Comportamento</i>	<i>66</i>
<i>Diagrama de Atividades</i>	<i>66</i>
<i>Diagrama de Casos de Uso</i>	<i>68</i>
<i>Diagrama de Máquinas de Estados</i>	<i>69</i>
<i>Diagramas de Interação</i>	<i>70</i>
<i>Diagramas de Sequência</i>	<i>71</i>
<i>Diagramas de Comunicação</i>	<i>72</i>
<i>Diagramas de Visão Geral da Interação</i>	<i>73</i>
<i>Diagramas de Sincronização</i>	<i>74</i>
CONCLUSÃO SOBRE UML	75
BPMN	76
<i>Elementos Básicos</i>	<i>77</i>
<i>Eventos</i>	<i>77</i>
<i>Atividades</i>	<i>78</i>
<i>Gateway</i>	<i>79</i>
<i>Fluxos</i>	<i>80</i>
<i>Dados</i>	<i>81</i>
<i>Artefatos</i>	<i>81</i>
<i>Colaborações</i>	<i>82</i>
CONCLUSÕES	84
REFERÊNCIAS BIBLIOGRÁFICAS	88
<i>Sites</i>	<i>91</i>
<i>Referências</i>	<i>92</i>

Prólogo

Este trabalho está diretamente ligado à forma como ocorreu o meu direcionamento profissional, desde a escolha da graduação até o momento em que de fato decidi atuar na área de tecnologia. Para explicar o que me motivou a pesquisar este tema preciso antes explicar um pouco da minha história para contextualizar o projeto.

Minha escolha pela graduação em Ciência da Computação teve uma motivação nem um pouco romântica. Optei por este curso porque era uma das carreiras mais promissoras e com possibilidade de maiores ganhos. Assim, fui direcionada à uma área que não era exatamente a que mais me encantava, visto que, minha primeira opção era jornalismo. Desde o início do curso, percebi que eu tinha um perfil diferente dos outros alunos, que se encantavam com a programação enquanto eu estava me divertindo com matérias mais teóricas, como Cálculo e Comunicação e Expressão Verbal. Ainda na faculdade me encantei com matérias como Banco de Dados e, principalmente, Programação Comercial, onde precisamos criar um projeto, desenhar o escopo e documentar todas as etapas – coisas que nenhum dos meus colegas tinha muito interesse em fazer.

Após meu primeiro emprego, escrevendo manuais de software e um breve período na área de desenvolvimento de uma multinacional brasileira, passei a trabalhar na área de Tecnologia de em uma agência de marketing onde eu: atendia o cliente, desenvolvia sistemas, criava o banco de dados, efetuava processamento de informações para geração de rankings e premiações das campanhas. Um trabalho de grande responsabilidade onde eu precisava cuidar de todas as etapas do processo, e qualquer erro poderia implicar em grande prejuízo financeiro para a empresa onde eu trabalhava. Felizmente, minhas habilidades de comunicação me ajudaram a me destacar e a perceber que, muitas vezes, os problemas que ocorriam nas entregas dos projetos estavam relacionados a uma falha na comunicação entre as pessoas envolvidas. Desde então, a questão da comunicação dos profissionais da área de Tecnologia da Informação tem me intrigado. Atualmente, como empresária e sócia de uma agência de marketing e uma empresa de tecnologia, que atuam juntas para fornecer soluções de gestão de informação e desenvolvimento de softwares web e aplicativos, principalmente para a área de Trade Marketing (marketing de varejo) de

grandes multinacionais, percebo que as falhas nos projetos por falta de comunicação afetam todo o desempenho da empresa.

Isso me motivou a estudar o problema e entender quais seriam as possíveis soluções. Mas para minha surpresa, percebi que é um tema reconhecidamente importante, porém pouco estudado ou retratado. Embora existam algumas ferramentas que tentam dar conta de resolver a questão da entrega e de melhorar os processos de desenvolvimento, quase não se fala do problema da comunicação em si. E este foi o fator que me levou a escrever este trabalho.

Introdução

Há uma concordância geral entre profissionais de todas as áreas no que diz respeito à importância da comunicação no âmbito profissional.

Entretanto, quando se trata de desenvolvimento de software, existe uma tendência em focar no desenvolvimento em si. Os profissionais buscam sempre se atualizar em relação as linguagens de programação mais modernas, conhecer e utilizar as melhores ferramentas do mercado para criar projetos e acompanhar sua evolução. Tudo isso é excelente, faz com que tenhamos capacidade de desenvolver soluções incríveis. Ainda assim, existe uma quantidade muito grande de projetos que não são entregues dentro do esperado e de sites e aplicativos que são desenvolvidos e nunca utilizados. Mas, se mapeamos os fluxos, estabelecemos processos e seguimos o planejado, porque ainda existem problemas nas entregas?

Todo fluxo e todo processo exigem que sejam seguidos sempre os mesmos passos. Quando os passos são operacionais e determinam ações a serem realizadas, é fácil seguir um padrão. Mas quando falamos em ações mais subjetivas, como comunicação, compreensão e percepção, não há como se desfazer do fator humano, muito mais difícil de padronizar e compreender.

A comunicação de cada indivíduo depende da forma como este percebe o mundo, de como percebe o outro, e ainda, de como ocorre a troca de informações ou de percepções entre dois indivíduos diferentes, para que ocorra uma comunicação efetiva. Para entender este problema precisamos então, entender as pessoas. Qual é o perfil dos profissionais, como eles interagem e como se comunicam para chegar ao entendimento do que deve ser realizado por cada um e por toda a equipe envolvida no projeto.

Estado da Arte

O desenvolvimento de um software exige uma capacidade cognitiva lógica focada na resolução de problemas. O foco é essencial para compreender e elaborar soluções para problemas complexos e chegar a um desfecho satisfatório. No

ambiente de desenvolvimento de software, um desfecho satisfatório implica em entregar um programa que resolva o problema proposto, no menor tempo possível, com o melhor resultado obtido e menor custo. Além disso o usuário deste software deve poder utilizá-lo naturalmente, sem esforço, tendo a percepção de simplicidade.

A formação acadêmica dos profissionais foi dirigida para a especialização das habilidades lógicas, de raciocínio matemático e desenvolvimento de habilidades que favoreçam o desenvolvimento de software, independentemente da especificidade do software. Entretanto, a abrangência da atuação do profissional deve extrapolar os limites técnicos, pois em 2017 os softwares estão presentes em atividades cotidianas em que sequer se questiona a existência de um programa.

Atualmente, exige-se que um profissional de Tecnologia da Informação seja altamente competente no que cabe às suas habilidades técnicas. Embora a comunicação seja uma ferramenta essencial a esses profissionais, existe pouco esforço no desenvolvimento desta habilidade, tanto no âmbito corporativo quanto acadêmico. As poucas pesquisas referentes ao desenvolvimento da comunicação para os profissionais de TI concentram-se na área de gestão de projetos, entretanto nem todos os profissionais são ou pretendem tornarem-se gestores, ficando, neste segundo caso, à deriva da bibliografia proposta para o desenvolvimento da comunicação. Quando tratamos de pequenas e médias empresas, que são o ambiente que estudaremos, esta habilidade é ainda mais importante considerando que cada profissional precisa assumir diversos papéis, distintos, no projeto. De acordo com GAVOLI (2009, p.40):

“Um indivíduo deve compatibilizar o elenco de habilidades adquiridas com a sua pretensão profissional na busca de uma qualificação, que ao mesmo tempo forneça o nível de formação exigido pelo mercado de trabalho e a conformidade com os valores éticos instituídos e as técnicas consagradas para a área de atuação pretendida”.

No entendimento de que as habilidades essenciais à ótima formação passam apenas por assuntos lógicos ou técnicos, o futuro profissional, ainda estudante busca sua formação na bibliografia técnica especializada. Profissionais da área de tecnologia

da informação comumente se tornam autodidatas, pois existe uma quantidade muito expressiva de documentos e cursos disponíveis na internet, o que facilita ainda mais o aprimoramento profissional.

Na busca pelo entendimento do real problema a ser solucionado, e visando convergir sua solução para problemas ainda não elaborados ou prever situações futuras decorrentes da solução proposta, o profissional de TI deve enriquecer seu repertório referente ao assunto objeto de seu projeto atual, bem como seus relacionamentos e derivações. E a ampliação de seu repertório passará certamente por extrair de seu interlocutor o contexto geral do cotidiano deste.

“Na longa história da cultura humana, a preocupação com os fenômenos da comunicação é uma preocupação recente. Ela data de meados do século XX, tendo coincidido com a explosão dos meios de comunicação de massa e a consequente emergência da cultura de massa” (SANTAELLA, 2001, p. 17).

Não surpreende então, que a preocupação com o desenvolvimento da habilidade da comunicação para profissionais de uma área tão técnica seja embrionária e praticamente inexistente. Na década de 80, o psicólogo Howard Gardner desenvolveu a Teoria das Inteligências Múltiplas, na qual sustenta que os indivíduos possuem diversos tipos de inteligências, e que é a combinação de nossas inteligências, que nos faz únicos e capazes de resolver problemas diversos. GARDNER (1995, p.18) diz:

“É de máxima importância reconhecer e estimular todas as variadas inteligências humanas e todas as combinações de inteligências. Nós todos somos tão diferentes em parte porque possuímos diferentes combinações de inteligências. Se reconhecermos isso, penso que teremos pelo menos uma chance melhor de lidar adequadamente com os muitos problemas que enfrentamos neste mundo”.

Justificativas

A tecnologia de software nos cerca a cada momento de nossas vidas. É raro um momento em que vivemos sem utilizar, direta ou indiretamente, algum produto ou serviço projetado sem a utilização de software, ou ainda que não o utilize para manter seu funcionamento. Diversos exemplos de serviços essenciais podem ser rapidamente citados, como controle financeiro, controle de reservatórios de água e rede elétrica, monitoramento de pacientes internados em UTI de hospitais, entre outros. Todos esses serviços em algum momento foram analisados e programados por profissionais de TI.

A tendência é que o controle informatizado aumente continuamente e que, ainda na primeira metade do século XXI, nossas casas, carros e até mesmo eletrodomésticos tornem-se totalmente controlados por software. O que nos leva a uma questão: “O que acontecerá se houver erros nos projetos que controlam os serviços básicos do nosso cotidiano, uma vez que, após automatizados e incorporados, dificilmente retornaremos ao mundo analógico?”. Pensar na importância da qualidade dos serviços desenvolvidos pelo profissional de TI é pensar na qualidade de vida e segurança de todas as pessoas.

Objetivos e metodologia

O objetivo deste projeto é pesquisar a importância da comunicação para o desempenho dos profissionais de Tecnologia da Informação e como isso pode afetar tanto positiva quanto negativamente o desempenho deste profissional.

Para tanto, vamos entender como Merleau-Ponty através de suas teorias sobre percepção pode contribuir para o entendimento da troca de informações que deve ser o resultado da comunicação efetiva, e como Martin Buber, através de suas teorias sobre o diálogo e o olhar sobre o Eu e o Tu podem nos ajudar a compreender verdadeiramente o outro. A falta de bibliografia referente ao problema da comunicação dos profissionais de TI fez com que houvesse grande dificuldade na elaboração do

mesmo. Em função disso, foi necessário ampliar o espectro e pensar na comunicação como um todo, buscando nesses grandes filósofos apoio para o entendimento deste tema tão complexo.

Serão apresentadas também duas ferramentas de modelagem, UML e BPMN, técnicas que buscam, na medida do que é tratável por um processo formal, tratar o problema da comunicação no desenvolvimento de software, através do desenvolvimento de diagramas que permitem detalhar todo o escopo do projeto, assim como as atividades a serem realizadas, e sequência correta a ser executada em cada ação.

Capítulos

Os dois primeiros capítulos foram difíceis de realização, porque o impacto dos problemas da comunicação no desenvolvimento de software não parece ser um tema tratado nos livros e manuais que cobrem o processo. Isso provocou o desafio de escrever uma descrição do problema dispondo de poucas referências.

No capítulo 1 serão apresentados quais são os cargos mais comuns entre os profissionais envolvidos no desenvolvimento de software, e como se dão as relações entre eles, e entre o interlocutor da equipe e o cliente, no processo de entendimento do projeto que deve ser entregue. Será apresentado um cenário das pequenas e médias empresas no Brasil atual para facilitar a compreensão do perfil que devem ter os profissionais que atuam nessas empresas e os desafios que irão enfrentar.

Mostraremos a evolução das profissões que deu origem à segmentação que ocasionou a necessidade de interação entre diversos profissionais, além de apresentar, de modo sintético, a teoria das múltiplas inteligências, como apoio para entender o perfil dos profissionais desta área e a forma como se adquirem e transmitem habilidades e conhecimento, outro ponto crucial no desenvolvimento de projetos bem-sucedidos, bem como a teoria da gestão de conhecimento, fundamental para a transmissão do conhecimento dos profissionais mais experientes para os demais. Ainda neste capítulo será apresentado um estudo de caso, para exemplificar os problemas que as falhas na comunicação podem causar em um projeto, gerando retrabalho e impactando nos custos.

No capítulo 2 falaremos do diálogo a partir do ponto de vista de dois grandes filósofos que trataram do tema: Maurice Merleau-Ponty e Martin Buber. Merleau-Ponty trata da percepção do indivíduo e de como a percepção de si próprio e do outro afeta a comunicação; Buber trata do diálogo e das relações entre o Eu e o Tu. Trata-se de um mergulho num terreno de pensamento pouco frequentado pelos profissionais de TI, mas que, por isso mesmo, nos parece extremamente importante para iluminar aspectos processuais historicamente negligenciados pelos modelos de gestão de projetos nesta área.

Já no capítulo 3 serão apresentadas duas ferramentas para modelagem de processos criados com o objetivo de explicitar, documentar e estabelecer processos padronizados que otimizam a compreensão e execução de tarefas. Vamos mostrar os diagramas básicos dos dois tipos de modelagem, no que cabe à modelagem de regras de negócios, a documentação de sequências em que as tarefas devem ser executadas, a interação entre elementos, exceções, transmissão de dados e mensagens e demais elementos utilizados no desenvolvimento dos diagramas de UML e BPMN.

Finalizando, o capítulo 4 apresentará os dados conclusivos do trabalho.

Cenário atual do profissional de desenvolvimento de software

As profissões relacionadas ao desenvolvimento de software são profissões relativamente novas na história da humanidade, visto que os primeiros computadores surgiram na década de 1940, e os primeiros computadores pessoais surgiram na década de 1970.

Por muitos anos os profissionais habilitados a operar e programar essas máquinas necessitavam de um alto nível de habilidades técnicas e matemáticas, dada a grande complexidade dos aparelhos.

À medida que a tecnologia se difundiu, custos foram reduzidos e mais pessoas passaram a ter acesso a computadores pessoais, tablets e telefones celulares, aparelhos físicos que não funcionam sem software. Dessa forma, cresceu proporcionalmente a necessidade de profissionais capacitados a desenvolver softwares nas mais diversas plataformas, atendendo à demanda crescente por ferramentas funcionais, informativas e de entretenimento.

As universidades também se adequaram, criando desde cursos de conhecimento amplo como Ciência da Computação, até cursos técnicos de especialização como Processamento de Dados. À medida que a demanda por software aumentou, outros cursos surgiram com o objetivo de atender subáreas específicas deste campo de conhecimento.

O modelo de negócios existente para a comercialização de softwares também sofreu alterações. O número de empresas de desenvolvimento de software aumentou consideravelmente, assim como os investimentos na área. Apenas em 2015, mesmo num contexto de grande recessão econômica, o desenvolvimento de software sob encomenda cresceu 9,4% em relação ao ano anterior, obtendo aproximadamente 1,4 bilhões em investimentos no Brasil, de acordo com o estudo Mercado Brasileiro de Software e Serviços 2016, produzido pela Associação Brasileira das Empresas de Software (ABES) em parceria com a IDC (International Data Corporation).

SEGMENTAÇÃO DO MERCADO BRASILEIRO DE SERVIÇOS / <i>Software Brazilian Market Segmentation</i>			
Segmento <i>Segment</i>	Volume <i>Volume</i>	Participação <i>Share</i>	2015 / 2014
Outsourcing / <i>Outsourcing</i>	6.075	40,5%	+ 9,6%
Serviços de Suporte / <i>Support Services</i>	2.709	18,1%	+ 5,2%
Integração de Sistemas / <i>System Integration</i>	2.242	15%	+ 7,5%
Consultoria e Planejamento / <i>Consultancy & Planning</i>	1.494	10%	+ 7,6%
Software sob Encomenda / <i>Taylor Made Software</i>	1.404	9,4%	+ 10,8%
Treinamento / <i>Training</i>	279	1,9%	+ 1,9%
Serviços para Exportação / <i>Services Export</i>	680	4,5%	+ 28,2%
Desenvolvido no Exterior / <i>Foreign Development</i>	97	0,6%	+ 26,6%
Subtotal Serviços / <i>Services Subtotal</i>	14.980	100%	+ 9%

Figura 1 – Segmentação do mercado brasileiro de serviços.

Fonte:

<http://central.abessoftware.com.br/Content/UploadedFiles/Arquivos/Dados%202011/ABES-Publicacao-Mercado-2016.pdf>. Acessado em: 08/06/2017

Um maior número em investimentos aumenta também a visibilidade do setor, direcionando mais profissionais à área de TI e também levando ao surgimento de um maior número de empresas atuando no setor.

É importante ressaltar que “setor de TI” é algo muito vasto e abrange uma grande variedade de tipos e tamanhos de empresas. O tipo de trabalho e o perfil de profissionais que atuam nessas empresas também é muito variado e seria muito complexo tratar do profissional de TI de forma tão ampla. Dessa forma, neste trabalho, trataremos apenas dos profissionais envolvidos no processo de desenvolvimento de softwares web, tais como sites e sistemas online, e aplicativos para aparelhos celulares em pequenas e médias empresas de desenvolvimento de software sob encomenda, um segmento que tem crescido de forma considerável nos últimos anos.

Pequenas e médias empresas

A pesquisa *As PMEs que mais crescem no Brasil* realizada em 2014 pela consultoria Deloitte em parceria com a PME (Pequenas e Médias Empresas, da revista Exame), concluiu que o setor de Tecnologia da Informação foi o que mais cresceu entre as empresas de pequeno e médio porte no período de 2011 a 2013.

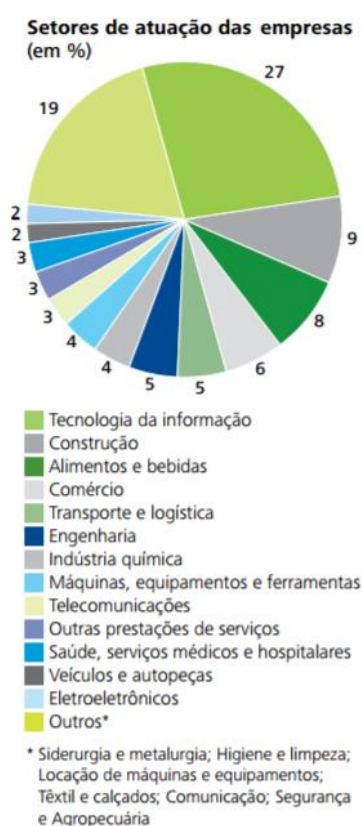


Figura 2 – Setores de atuação das empresas em percentuais.

Fonte:

<https://www2.deloitte.com/content/dam/Deloitte/br/Documents/conteudos/pmes/PME2014.pdf>

f. Acessado em: 08/06/2017

As empresas que comercializam software desenvolvido sob encomenda possuem um grande desafio a cada projeto, pois seus clientes podem escolher todas as funcionalidades que a aplicação terá ao final do desenvolvimento e ao mesmo

tempo é um público que pode não saber exatamente quais funcionalidades necessita, levando à solicitação de alterações consideráveis no meio do projeto. O cliente muitas vezes não percebe o impacto das alterações solicitadas e não entende quando elas não podem ser realizadas, já certo nível de mudanças solicitadas durante o desenvolvimento torna um projeto inviável para a empresa desenvolvedora, do ponto de vista comercial. A expectativa do cliente é de que o software seja funcional, estável, visualmente atrativo, e, sobretudo produzido com baixo custo e em prazo mínimo.

Os profissionais que atuam neste modelo de negócios precisam entender e atuar em diversas etapas do projeto que vão desde escrever o código fonte até desenhar telas de forma visual, arquitetar bancos de dados que permitam maior agilidade na busca por informações, mapear e conectar informações armazenadas. Enfim, o profissional da área de tecnologia da informação - ou profissional de TI como é conhecido precisa, frequentemente exercer múltiplos papéis e atuar em diversas fases do desenvolvimento, interagindo com seus parceiros de equipe de forma muito próxima, e por vezes sem uma clara definição do responsável por cada etapa a ser concluída.

Em função dessa interação entre as pessoas da equipe, o profissional de TI mais do que nunca passou a precisar se comunicar bem. A comunicação é essencial para o ser humano, pois é através dela que ocorre a troca com o outro. Podemos falar ou escrever, o essencial é que o outro receba e decodifique a mensagem enviada, assim como acontece no software.

Maurice Merleau-Ponty inicia seu livro *A Prosa do Mundo* dizendo: “Na Terra, já se fala há muito tempo, e a maior parte do que se diz passa despercebido” (MERLEAU-PONTY, 2012, p.29). É comum, ao assistirmos uma palestra ou mesmo em uma conversa informal, escutarmos o que o outro diz, sem compreender boa parte daquilo que ele desejou comunicar, por falta de atenção ou porque as palavras não nos cativaram. MERLEAU-PONTY (2012, p. 196) diz: “Quando tenho a impressão de me ocupar apenas com palavras, é que a expressão falhou; Ao contrário, se ela é bem sucedida, parece-me que penso ali, em voz alta, naquelas palavras que eu não disse”.

Um interlocutor sem conhecimento técnico, ao ouvir uma sequência de termos técnicos que não fazem sentido para ele, corre o risco de ater-se às palavras e perder o sentido amplo daquilo que foi dito. Cabe ao palestrante, escolher as palavras com

cuidado, navegando no oceano de seu interlocutor e atribuindo sentido às palavras ditas. Isso se faz ainda mais importante nas relações comerciais, onde o interlocutor, ou cliente, pode não ter a intenção de se informar sobre os termos técnicos e ainda assim, precisa compreender o que lhe é dito.

A comunicação então é fundamental para que essas relações entre pessoas de uma equipe e cliente-empresa ocorram de forma a garantir a satisfação do cliente, visto que ele é o responsável pela continuidade e sucesso da empresa.

Otimização de recursos

No Brasil, os altos encargos pagos pelas empresas e a necessidade de todos os cidadãos, inclusive empresários, de pagar por necessidades básicas como saúde e educação fazem com que os pequenos e médios empresários precisem otimizar ao máximos seus recursos. Para essas empresas reduzir custos não significa apenas aumentar lucro, significa sobreviver. A Pesquisa do SEBRAE intitulada Sobrevivência das Empresas no Brasil, realizada em outubro de 2016, indica que 34% das empresas de Desenvolvimento de programas de computador sob encomenda fecham antes dos dois primeiros anos. Este cenário, leva os empresários a uma corrida por entregar projetos com a melhor relação custo-benefício possível e, para que esta entrega seja feita, a equipe envolvida no desenvolvimento dos sistemas acaba sendo afetada, no sentido de que precisa realizar a maior quantidade de trabalho possível, no menor tempo, sem abrir mão da qualidade da entrega.

Uma das formas de facilitar o alcance deste objetivo, sem sobrecarregar os colaboradores, é melhorar os processos de desenvolvimento, muitas vezes menos formalizados nas empresas de pequeno e médio porte. O primeiro passo para a formalização dos processos é entender o papel de cada pessoa no desenvolvimento do software, pois, com a evolução da área de tecnologia da informação surgiram especialistas em todas as áreas, tais como programação, banco de dados, design e outros.

Essa forma especialista e focada de organizar o pensamento surgiu no final do século XVIII. Antes disso, as crianças eram preparadas para aprender todas as fases

do processo de construção de uma casa, por exemplo, desde a mais tenra idade. No livro *Child labour in historical perspective* (CUNNINGHAM e VIAZZO, 1996) (Trabalho infantil em perspectiva histórica, tradução livre) Hugh Cunningham e Pier Paolo Viazzo citam que os dados estatísticos são muito falhos para estimar a idade em que as crianças começavam a trabalhar, porque era prática comum as crianças ajudarem no sustento de suas famílias, embora acredita-se fosse um trabalho não-exploratório. Dessa forma, desde muito jovens e até o fim de suas vidas formavam-se alfaiates, sapateiros, construtores e diversas outras profissões. Todo o processo de aprendizagem de formação profissional implicava em conhecer todas as fases do trabalho.

Em 1794, Eli Whitney patenteou uma máquina criada por ele para automatizar o processo de separação das sementes de algodão das fibras: “O dispositivo produzia mais algodão em uma hora do que o que poderia ser produzido por vários trabalhadores em um dia”¹. Mesmo com a patente, Whitney não conseguiu obter lucros, pois os fazendeiros passaram a produzir suas próprias máquinas descaroçadoras de algodão. Por conta de suas dificuldades em receber os lucros pela descaroçadora de algodão Whitney aceitou o desafio de desenvolver dez mil armas para o exército em dois anos. Até então as armas eram produzidas individualmente por artesãos “Na época, os mosquetes eram geralmente montados na sua totalidade por artesãos individuais, com cada arma tendo seu próprio design distinto”².

Whitney desenhou moldes, para que todas as peças fossem produzidas no mesmo padrão para serem posteriormente montadas de forma idêntica: “Quando juntas, cada parte, embora feitas separadamente, tornavam-se um modelo que funcionava”³. Dessa forma, Whitney reduziu expressivamente o tempo de produção. Embora o primeiro lote de dez mil armas tenha levado oito anos a mais do que o previsto, o segundo lote solicitado, que era de quinze mil unidades, foi concluído em dois anos.

Este sistema de peças intercambiáveis foi altamente utilizado durante a revolução industrial onde os trabalhadores eram responsáveis por apenas um item específico da linha de montagem, e, aos poucos, perdeu-se a necessidade do conhecimento abrangente. Dessa forma, o processo de formação dos profissionais também foi alterado. O empregador, que antes era também o responsável pelo treinamento dos funcionários, passou a ser apenas um receptor do trabalho com a

contrapartida de pagar um salário ao trabalhador. E o conhecimento começou a ser especialista. Quanto mais especialista fosse o trabalhador, mais valorizado ele era.

No final do século dezenove, Frederick Taylor criou a teoria do Gerenciamento Científico, onde analisou todos os passos do processo de produção, e dividiu cada passo em uma tarefa simples que trabalhadores pouco qualificados poderiam realizar sem dificuldades. Esses trabalhadores com menos habilidades eram aqueles que poderiam executar tarefas repetitivas sem necessidade de raciocinar sobre a tarefa. Eles também eram facilmente substituíveis, principalmente em função da grande oferta de mão-de-obra gerada em decorrência da intensa migração da população das áreas rurais para os centros urbanos: “em 1790, setenta e cinco por cento da força de trabalho consistia de trabalhadores rurais, e esta taxa caiu para quarenta por cento perto da virada para o século vinte”⁴. Dessa forma, a produção aumentou consideravelmente, gerando demanda por novas máquinas, e consequentemente, por profissionais qualificados para projetar, construir e dar manutenção nos equipamentos.

Com o tempo, as tarefas dos profissionais mais qualificados também foram segmentadas para melhorar a produtividade dessas tarefas. Assim, o engenheiro foi treinado para ser responsável apenas pelo motor da máquina, enquanto outro ficou responsável pela montagem.

Essa nova forma de nos posicionarmos em relação ao trabalho correspondeu a uma grande mudança em nossa forma de pensar, descrito por Foucault como “sociedade disciplinar”⁵. Separamos as ciências exatas, das ciências humanas e biológicas. E mesmo dentro dessas áreas fragmentamos em diversas subáreas. A engenharia adquiriu uma extensa variedade de subdivisões; a medicina, uma das áreas onde se torna mais evidente a alta demanda por especialidade dos profissionais, aprimorou a atuação reducionista de seus profissionais a tal ponto que tornou-se pouco comum encontrar médicos que saibam olhar o todo do paciente. Todos conhecem histórias, por exemplo de um ortopedista que trata de uma dor no pé como um problema ortopédico, para posteriormente o paciente descobrir que as dores eram causadas por um problema neurológico.

Até mesmo nas áreas de desenvolvimento de software, como dito anteriormente, novas profissões de tecnologia surgiram. Existem especialistas em

programação, arquitetura de software, banco de dados, web designer, atendimento ao cliente, e outros. Cada profissional é responsável por uma parte fundamental do desenvolvimento do software e a especialidade se tornou tão específica que foi necessário surgir um novo profissional para facilitar a interação entre todas as partes do desenvolvimento. Surgiu assim o Gerente de Projeto. Este profissional tem o objetivo de gerenciar as entregas das tarefas dos diversos profissionais e garantir que haja certa harmonia entre as ideias: tal qual um maestro deve liderar sua orquestra, conduzindo os diversos talentos musicais para soarem harmoniosamente, o gerente do projeto deve conduzir os profissionais do projeto para uma entrega que atenda à expectativa do cliente no prazo correto e dentro do custo esperado.

Mas na condução de um projeto algumas diretrizes devem ser seguidas. O que define as diretrizes é o método escolhido. Embora exista uma grande quantidade de metodologias de gestão de projetos e desenvolvimento de software, nem todas ajudam a obter sucesso: “As regras de gerenciamento e administração atualmente em uso foram desenvolvidas e estão organizadas para mercados e tecnologias estáveis ou que se alteram em processos graduais e lentos” (OLIVEIRA, 1996, p.15).

Precisamos olhar para as metodologias, para a formação e para as diretrizes de forma mutável, assim como acontece com o próprio desenvolvimento de software. O mundo moderno não pode, em hipótese alguma, ser considerado estático. Inúmeras forças impõem novas concepções à sociedade e às empresas, sendo a rapidez no ritmo das mudanças a mais importante delas. Os avanços tecnológicos atuais estão obrigando empresas, produtos e serviços a mudanças e adaptações em velocidade sem precedentes. Estratégias empresariais que parecem interessantes um dia revelam-se obsoletas no dia seguinte. (OLIVEIRA, 1996, p.14)

É de conhecimento de todos que há problemas na entrega de projetos de desenvolvimento de software. Em algum momento, todos já ouviram a famosa frase: “O sistema está fora do ar”. Nunca sabemos exatamente o que originou esta queda no sistema, pois existem diversos motivos que podem gerar uma falha, tais como, problemas com a conexão de internet ou erro de programação. Entretanto, sabemos que existem problemas com as entregas de software, porque nos deparamos muitas

vezes com erros de funcionamento dos sistemas. O grupo internacional Tha Standish Group publica, anualmente, um relatório que apresenta os índices de falhas nos projetos de desenvolvimento de software, com o objetivo de ajudar os profissionais a identificar e corrigir os principais motivos que levam a falhas. Esse relatório se chama Chaos Report.

Chaos Report

O Chaos Report é um relatório desenvolvido desde 1994 para medir o sucesso no desenvolvimento de software.

MODERN RESOLUTION FOR ALL PROJECTS					
	2011	2012	2013	2014	2015
SUCCESSFUL	29%	27%	31%	28%	29%
CHALLENGED	49%	56%	50%	55%	52%
FAILED	22%	17%	19%	17%	19%

The Modern Resolution (OnTime, OnBudget, with a satisfactory result) of all software projects from FY2011–2015 within the new CHAOS database. Please note that for the rest of this report CHAOS Resolution will refer to the Modern Resolution definition not the Traditional Resolution definition.

Figura 3 - Resultados dos projetos

Fonte: <https://www.infoq.com/articles/standish-chaos-2015>. Acessado em 10/07/2016.

O índice foi avaliado também de acordo com o tamanho dos projetos.

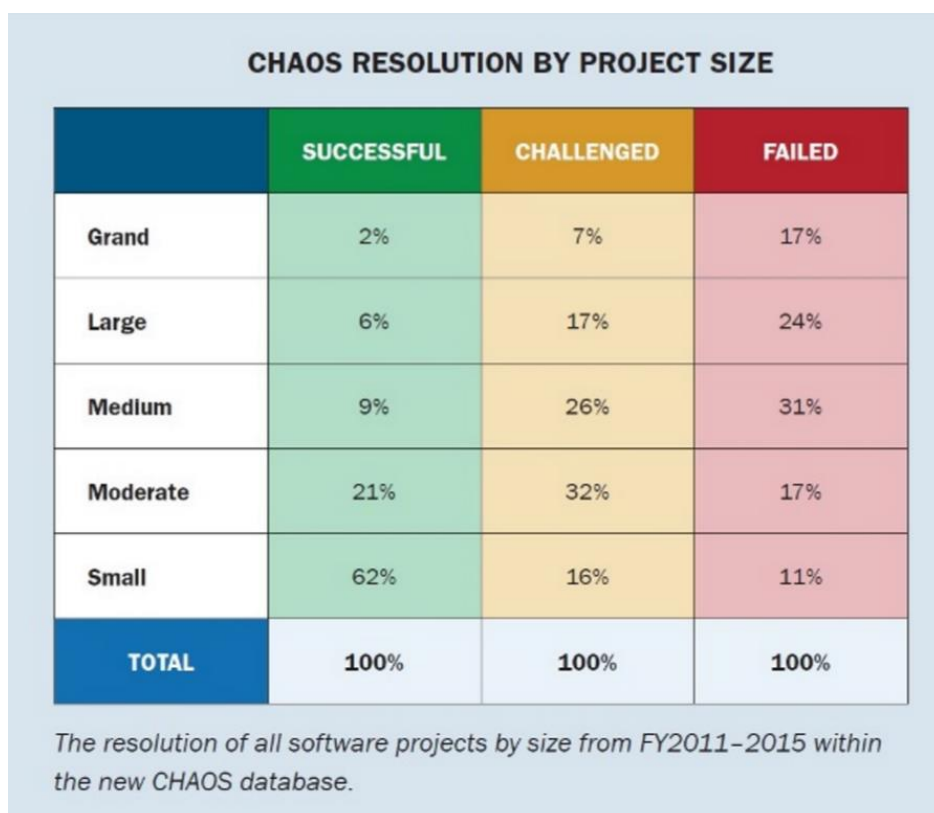


Figura 4 - Resultados dos projetos de acordo com o tamanho do projeto.

Fonte: <https://www.infoq.com/articles/standish-chaos-2015>. Acessado em 10/07/2016.

Através de um conjunto de medidas, indica o índice de sucesso dos projetos de desenvolvimento de software, considerando um projeto bem-sucedido aquele que é entregue dentro do prazo, dentro do custo e com um resultado satisfatório.

Diversos fatores são mensurados para atingir esses índices, cada um com seu percentual de contribuição, entre eles estão:

- Suporte executivo (15%);
- Maturidade emocional (15%);
- Envolvimento do usuário (15%);
- Otimização (15%);
- Equipe capacitada (10%);

- SAME (Standard Architectural Management Environment) ou ambiente gerenciado por uma arquitetura padrão (8%);
- Utilização de metodologia de desenvolvimento ágil (7%);
- Execução simplificada (6%);
- Expertise em gestão de projetos (5%);
- Alinhamento claro dos objetivos do projeto com os objetivos da Empresa (4%).

Segundo relatório, a *maturidade emocional* é responsável 15% do fator de sucesso dos projetos. Por maturidade emocional entende-se “a coleção básica de comportamentos de como as pessoas trabalham juntas”; ou ainda: “Em qualquer grupo, organização ou empresa é tanto a soma de suas habilidades e o elo mais fraco que determinam o nível de maturidade emocional”⁶ (CHAOS REPORT, 2015). O relatório aponta ainda que quanto maior o projeto, menor o índice de sucesso. Podemos deduzir que, quanto maior o projeto, maior é a quantidade de pessoas envolvidas nele, e, portanto, maior é a quantidade de interação entre pessoas, pessoas com percepções distintas e níveis de maturidade emocional distintos, e que precisam se comunicar de forma efetiva. Entretanto, pequenas e médias empresas que desenvolvem software sob encomenda, em geral, desenvolvem projetos pequenos o que aumenta o índice de sucesso na entrega, mas ainda assim aponta 11% de fracasso. Se faz importante, então, entender um pouco quem são os indivíduos que atuam no projeto. Qual é o papel de cada pessoa e como a equipe desempenha suas funções individuais para atingir o objetivo final comum. Vamos entender como aconteceu a evolução das carreiras até chegar nos cargos atuais, e qual é o perfil do profissional da área de Tecnologia da Informação atualmente.

Níveis de interação

Considerando projetos onde existe uma equipe, ou seja, mais de um indivíduo envolvido no desenvolvimento do software, é fundamental que exista um bom entendimento entre as pessoas da equipe. O problema é que a forma de pensar de cada indivíduo é altamente impactada por sua percepção do mundo. O ambiente onde foi criado, as influências que o indivíduo sofreu na escola e em todos os lugares por onde passou determinam sua forma de perceber cada coisa. Além disso, e este talvez

seja o ponto mais importante, mesmo que toda a equipe envolvida no desenvolvimento do software esteja com seus pensamentos alinhados, é fundamental que o trabalho da equipe esteja alinhado com o pensamento e a expectativa do cliente. “Todo sistema interessante interage com atores humanos ou autômatos que utilizam esse sistema para algum propósito e esses atores esperam que o sistema se comporte de acordo com as maneiras previstas” (BOOCH, 2012, p.246).

E este é o ponto crítico de todo o desenvolvimento de software: a expectativa do usuário.

As interações entre as pessoas ocorrem basicamente de duas formas distintas:

Relacionamento entre equipe de desenvolvimento e cliente

Normalmente existe um profissional da equipe responsável por ser o interlocutor. É para ele que o cliente irá ligar sempre que precisar de alguma coisa.

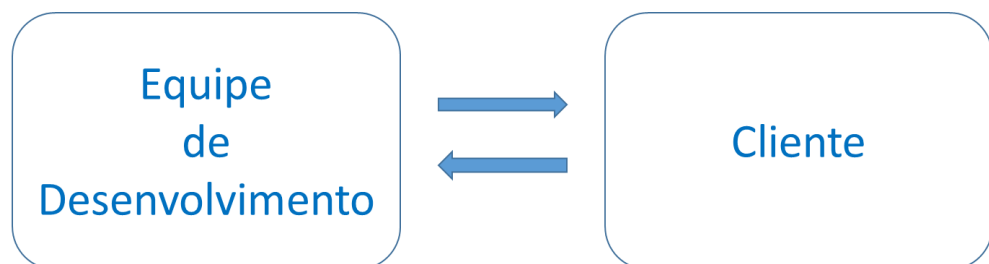


Figura 5 - Relacionamento Equipe-Cliente

Relacionamento entre as pessoas da equipe de desenvolvimento do software

Uma equipe são: duas ou mais pessoas reunidas com o objetivo de desenvolver e entregar um software para o cliente. Podem haver diversos níveis de interação entre essas pessoas. Em alguns casos há um nível hierárquico, mas em outros, são apenas colegas trabalhando em pares com um objetivo comum, por isso a figura 6 apresenta os profissionais que podem estar envolvidos no desenvolvimento de um software, mas não há indicação de fluxo de ligação entre os profissionais.

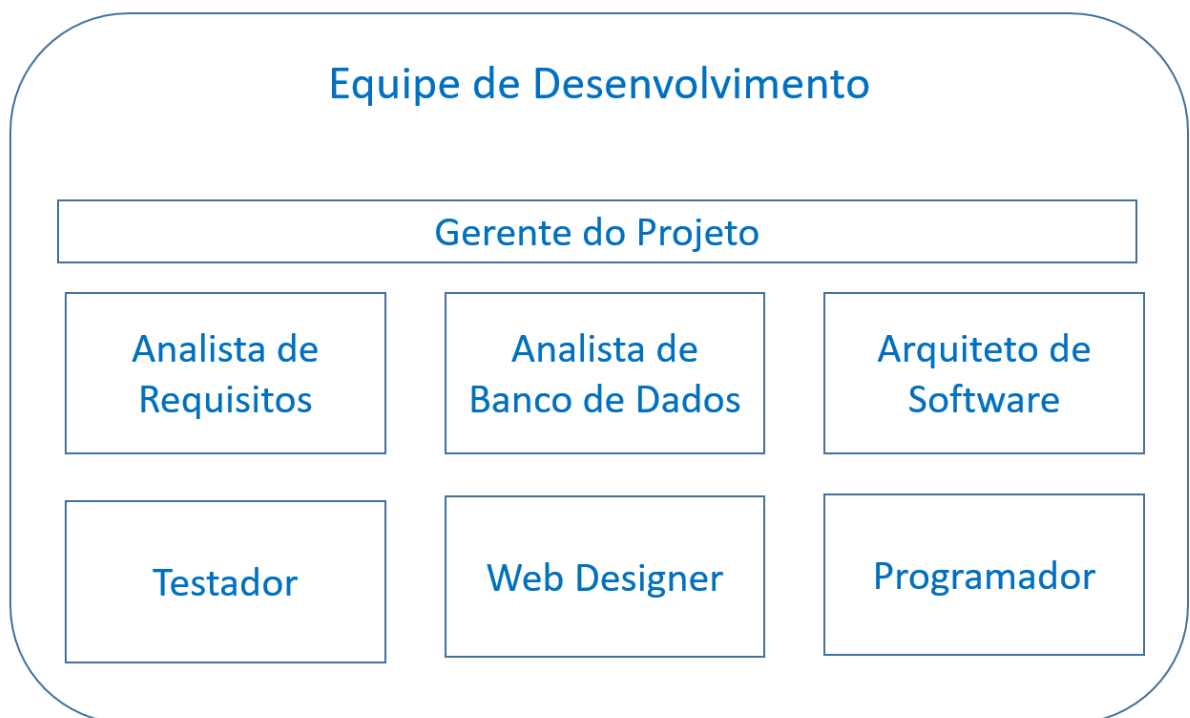


Figura 6 - Relacionamento interno da equipe

Uma equipe básica de desenvolvimento de ferramentas web é composta pelos seguintes profissionais:

- **Gerente do Projeto** – é o maestro da orquestra. É sua responsabilidade controlar prazos, custos, em alguns casos, coordenar a equipe, atender o cliente e apresentar resultados e andamento do projeto;
- **Analista de Requisitos** – Nem sempre existe uma pessoa com este cargo, muitas vezes o próprio gerente do projeto desempenha este papel. O analista de requisitos é o responsável por entender a expectativa do cliente e traduzir para a equipe. Talvez o papel mais importante considerando o escopo deste trabalho. Todos os membros da equipe deveriam em algum momento se colocar no papel de analista de requisitos;
- **Analista de Banco de Dados** – O analista de Banco de dados é o responsável pelo gerenciamento das informações do banco de dados. Sempre que uma função da ferramenta acessa os dados de um BD o analista é o responsável por gerenciar este acesso de entrada, processamento e saída de dados;
- **Arquiteto de Software** – O arquiteto tem o papel de dar o direcionamento aos programadores. Dentre outras coisas define o padrão de desenvolvimento que será utilizado, determina o aproveitamento de outras estruturas do sistema, muitas vezes criando bibliotecas de componentes que facilitam o trabalho dos programadores menos experientes;
- **Programador** – Cabe ao programador, escrever o código fonte por trás de cada função do projeto. Ele se relaciona diretamente com o analista de banco de dados e com o web designer, pois uma tela com funcionalidades pode ser construída por todos esses profissionais;
- **Redator** – O redator é responsável por criar e rever todos os textos da ferramenta. Em alguns casos o cliente envia os textos prontos, mas em geral é necessário que o redator revise para manter uma comunicação homogênea em toda ferramenta;

- **Web Designer** – web designer é o profissional responsável pela identidade visual da ferramenta. Ele define a tipografia, cores imagens e programa algumas funcionalidades mais superficiais como efeitos gráficos;
- **Testador** – é o responsável por efetuar testes na ferramenta e elaborar relatórios de falhas encontradas. O ideal é ter mais de uma pessoa testando para que não sejam executados testes viciados. Assim como o analista de requisitos, pode ser um papel efetuado por diversas pessoas da equipe, embora isso aumente o custo do projeto.

Quando reunimos diversas pessoas em um projeto e apresentamos um problema, cada uma formula o entendimento do problema do seu ponto de vista. Esse ponto de vista considera todo o repertório intelectual do indivíduo, suas experiências anteriores, suas preferências, etc.

Pensem em um exemplo de desenvolvimento de um produto qualquer: Se solicitarmos para um grupo de pessoas que façam uma cadeira, por exemplo. Imediatamente cada indivíduo irá formular uma imagem mental diferente em sua mente. Um poderá pensar em uma cadeira de madeira estofada com veludo, enquanto outro poderá visualizar uma cadeira de ferro dobrável e desconfortável. Se houver um sujeito que caiu de uma cadeira quando criança e se machucou, ele pode detestar a ideia de fazer este trabalho, outro que gostaria de ter estudado design pode adorar a ideia de realizar uma parte deste sonho. Assim a simples ideia do problema proposto desencadeia, não apenas imagens mentais, mas sentimentos favoráveis ou desfavoráveis ao problema apresentado. Todo gestor de equipe sabe que eventualmente indivíduos boicotam inconscientemente o projeto por não gostarem, não acreditarem ou por sentirem-se desconfortáveis com a ideia do projeto. Tudo depende da percepção a respeito da ideia proposta.

É necessário ainda levar em conta as habilidades naturais de cada indivíduo. Aqueles que possuem uma habilidade estética mais apurada, deveriam naturalmente ser direcionados para fases do projeto que cuidam da aparência do resultado a ser produzido pelo projeto e indivíduos com maior habilidade para cálculos deveriam ser direcionados para esta área. No mundo corporativo entretanto, a formação

especialista do indivíduo normalmente o direciona para uma área de atuação sem considerar essas habilidades que superam ou são paralelas à formação.

É muito comum, em reuniões onde o cliente (não técnico) encontra um interlocutor técnico, surgir uma situação extremamente desconfortável onde, de um lado, o sujeito técnico explica com toda sua calma os detalhes técnicos de andamento do projeto e o cliente não entende absolutamente nada do que está sendo dito. Em geral, o sujeito técnico quer demonstrar os problemas encontrados e não o impacto que isso terá na solução. É como se um mecânico explicasse todo funcionamento do motor de um carro para explicar que este parou de funcionar porque acabou a gasolina. Para o cliente em geral só interessam três coisas: “é possível fazer ou consertar?”, “quanto tempo leva?”, “quanto vai custar?”. Clientes com mais conhecimento técnico, ou mais curiosos, farão perguntas e essas sim poderão ser respondidas em mais detalhes, desde que haja receptividade.

Um outro problema muito comum é que alguns profissionais associam o uso de termos técnicos específicos e preferencialmente em inglês a um maior conhecimento técnico. Seja por vaidade ou por ignorância, esses profissionais acreditam que ao deixar o cliente sem entender o que dizem vai causar a impressão de que são muito capacitados a realizar o trabalho. Entretanto, os clientes tendem a ter mais empatia por profissionais que os ajudam a entender os termos que não conhecem, e que “traduzem” o problema para o idioma conhecido, sem tantos termos técnicos e difíceis de se pronunciar. Isso aumenta a confiança do cliente no fornecedor e aumenta também as chances de fechar negócios.

Ao tentar resolver os problemas de atendimento e interação com o cliente, alguns projetos são liderados por profissionais com vasta experiência em atendimento, mas sem conhecimentos de desenvolvimento de software, o que leva a outro problema. Uma reunião pode correr muito bem com o cliente muito satisfeito, mas sem algumas respostas importantes que a equipe de desenvolvimento precisa. Isso gera um dilema entre o interlocutor do cliente ser um profissional com perfil de atendimento, mas sem conhecimento técnico ou um profissional com grande conhecimento técnico, mas sem perfil de atendimento.

Estudo de caso

Vamos entender um caso onde a comunicação, ou a falta dela, gerou perda de tempo e retrabalho. Este é um caso real que aconteceu em minha empresa, porém, por contrato de confidencialidade, o nome da empresa não pôde ser revelado. Também o nome do profissional envolvido foi omitido a fim de evitar qualquer constrangimento para o mesmo.

Trata-se de uma plataforma corporativa criada para centralizar a comunicação de uma área da companhia. Com cerca de trinta perfis de acessos diferentes, cada usuário, ao efetuar o login, consegue visualizar as informações pertinentes ao seu perfil de acesso, de acordo com o cargo e as permissões de visualização que lhe foram conferidas pelos administradores da plataforma.

Através de uma das funcionalidades da plataforma, chamada *biblioteca*, os usuários podiam baixar arquivos como relatórios disponibilizados em planilhas de Excel e arquivos de texto com extensões *.docx* e *.pdf*. Os arquivos eram apresentados em formato de lista com a data de publicação, nome e tipo de arquivo, ordenados alfabeticamente.

A plataforma já estava em funcionamento há pouco mais de um ano quando o cliente identificou a necessidade de disponibilizar para alguns usuários, vídeos e apresentações em Power Point. Na reunião de *briefing*, onde o cliente explicou a demanda, estava presente o analista de banco de dados responsável pelo projeto, fazendo naquele momento o papel de analista de requisitos (vale lembrar que este estudo trata de pequenas e médias empresas onde nem sempre existe uma pessoa responsável por desempenhar cada papel no desenvolvimento do projeto). Muitas vezes, e é este o caso, o analista de banco de dados faz a interlocução com o cliente, recebe a demanda, especifica, envia para a programação, acompanha, testa e efetua a entrega.

A demanda foi recebida, o prazo acordado com certo pesar. O cliente não entendia porque precisariam de tanto tempo para fazer uma alteração tão simples.

O analista tentou explicar que não era algo tão simples como parecia, mas achou que o cliente não ia entender mesmo, então voltou ao escritório para trabalhar. Esboçou o layout da nova tela onde imagens com miniaturas de vídeos eram

apresentadas na tela. Para as apresentações seriam mostrados ícones com o logo do Power Point do mesmo tamanho das imagens em miniaturas dos vídeos. O mesmo aconteceria com os arquivos de texto.

Foi envolvido o web designer, para que pudesse desenhar a tela de forma bem visual, depois, criada a estrutura de banco de dados para que pudesse retornar o tipo de arquivo e a imagem que seria apresentada para cada um deles. Também havia no escopo, uma nova área administrativa que permitiria a publicação dos vídeos e o upload das imagens em miniatura que iriam ilustrar os arquivos.

Especificação pronta e detalhada com *procedures* (procedimentos) e outras funções do banco de dados criadas e testadas, o projeto chegou à programação. O programador responsável criou o ambiente de homologação para esta área. Como diversos programadores trabalhavam no projeto e a plataforma era bastante complexa, haviam combinado sempre atualizar o ambiente de homologação para as áreas que seriam alteradas pouco antes de iniciar o trabalho. Assim teriam sempre o último ambiente disponível.

Na programação tudo correu bem e o projeto foi concluído quase dentro do prazo. Com atraso de apenas alguns dias, com a compreensão do cliente. No dia da entrega o analista entrou na sala de reuniões e projetou seu trabalho com orgulho no telão, certo de que o cliente iria concordar que valeu a pena a espera. Mas, diferentemente do que ele esperava, um silêncio tomou conta da sala. O cliente olhava para a tela cheia de ícones sem dizer nada. Então o analista perguntou o que ele tinha achado e ele respondeu um pouco apreensivo que não era bem aquilo que ele havia pedido. Na verdade, ele só queria poder incluir, na lista de arquivos disponíveis para download, vídeos e apresentações. Eles não precisavam ser apresentados em miniaturas de imagens. Ele achava que daquela forma iriam aparecer menos arquivos por tela e poderia confundir os usuários já acostumados à lista de arquivos anterior.

Não é necessário falar sobre a frustração generalizada, sem contar a irritação dos demais profissionais quando perguntaram se o cliente tinha gostado do trabalho que eles tinham ficado concluindo até tarde, e que jamais seria utilizado. Alguém perguntou para o analista se ele não havia perguntado se era isso que o cliente realmente queria e ele respondeu que ele tinha certeza de que isso foi pedido.

O fato é que além da grande falha de processo, sem apresentar o protótipo para aprovação do cliente, o analista não conversou com o cliente para entender o que de fato ele precisava e porque queria esta alteração. Não tentou entender qual era o real objetivo da alteração solicitada. Mais quinze minutos de conversa talvez fossem suficientes para entender que o cliente tinha um arquivo de vídeo que queria disponibilizar para um usuário de outro Estado e quando tentou publicar, percebeu que a ferramenta não permitia o upload este tipo de arquivo.

O analista não perguntou porque estava claro para ele o que deveria ser feito. Ele havia entendido tudo e quis agradar o cliente e fazer um trabalho muito bem feito. E o fez. Assumiu inconscientemente que sua percepção era a única verdade e seguiu com ela. Não teve, entretanto, a sensibilidade de perceber o que o cliente realmente precisava ou a habilidade para se comunicar com o cliente de forma mais eficiente.

Dificuldades em torno do problema

As pessoas interagem de acordo com seus pontos de vista sobre o mundo. A educação e o caminho profissional que cada um segue ajuda a moldar essa visão, mas a essência de cada um é única e, é a base para a formação de cada profissional. Pessoas com profissões idênticas podem ter percepções completamente diferentes a respeito do trabalho, do chefe, enfim, do mundo.

Howard Gardner é um pesquisador norte-americano, da Universidade de Harvard, que publicou a Teoria das Inteligências Múltiplas onde contestou as teorias de Jean Piaget, que, segundo Gardner (1995, p3) “...via todo o pensamento humano como lutando pelo ideal do pensamento científico; e a prevalente concepção de inteligência que vinculava à capacidade de dar respostas sucintas, de modo rápido, a problemas que requerem habilidades linguísticas e lógicas”. Gardner acreditava que os seres humanos possuem “um número desconhecido de capacidades humanas diferenciadas, variando desde a inteligência musical até a inteligência envolvida no entendimento de si mesmo” (GARDNER, 1995, p3).

Na sua Teoria de Inteligências Múltiplas, Howard Gardner afirma que o ser humano é dotado de diversas inteligências e não apenas as inteligências lógico-matemática e linguística, responsáveis pela capacidade de soluções lógicas e de elaboração de textos, respectivamente. As demais capacidades, como, por exemplo a capacidade intrapessoal responsável pela interpretação e visão do outro, além da comunicação e trato com outros sujeitos, deveriam ser desenvolvidas pela formação acadêmica com o objetivo de proporcionar ao sujeito o pleno uso da maior quantidade de inteligências possível, “É da máxima importância estimular todas as mais variadas inteligências humanas e todas as combinações de inteligências” (GARDNER, 1995, p. 18).

Ao desenvolver todas as inteligências, mesmo que respeitando as aptidões e preferências do sujeito, que certamente irá ter melhor desempenho nas áreas que lhe são natas e potencializadas pela influência do ambiente em que ele cresceu, conseguiremos ampliar a visão macro dos problemas, para que o sujeito consiga estabelecer o foco sem perder a visão do todo: “Nós todos somos tão diferentes em

grande parte porque possuímos diferentes combinações de inteligências” (GARDNER, 1995, p. 18).

Por outro lado, se as diversas inteligências não forem desenvolvidas o sujeito ficará restrito a uma análise parcial de problemas, pois utilizando apenas a sua inteligência mais facilmente desenvolvida, seu raciocínio tende a ir sempre pelo mesmo caminho. Assim, o sujeito torna-se limitado em suas visões, e consequentemente, limitado em suas ações. Para o mercado de trabalho atual este é um dos principais defeitos de um profissional: não ter a capacidade de expandir seu pensamento e olhar um problema de diversos pontos de vista. O mercado de trabalho, cada vez mais, tem exigido um conhecimento mais abrangente sobre assuntos diversos, sem abrir mão de uma especialização em algum assunto específico. Os profissionais têm se destacado por ter experiências como trabalhos voluntários, viagens ao exterior, cursos em áreas completamente diferentes das áreas em que atuam. O pacote graduação + pós-graduação + segundo idioma, ainda é essencial, mas já não destaca nenhum profissional.

Quando se trata de profissionais de tecnologia, de maneira geral, encontramos uma grande quantidade de sujeitos dotados de *inteligência lógico-matemática* - uma das inteligências que, segundo Gardner, é utilizada para os testes de QI tradicionais, mas que na prática não garantem sucesso às pessoas: “Há idiotas sábios que realizam grande façanhas em cálculo, mesmo que continuem sendo tragicamente deficientes na maioria das outras áreas” (GARDNER, 1995, p. 25). Isso ocorre porque as pessoas com uma inteligência lógico-matemática possuem mais facilidade em relação as chamadas ciências exatas, das quais os cursos de formação de desenvolvedores de software fazem parte.

Outra inteligência tradicionalmente reconhecida pela psicologia é a *Inteligência linguística*, ou a capacidade de se comunicar com as pessoas. Muito comum, por exemplo, em jornalistas e advogados, está, sem dúvida, entre as maiores dificuldades dos profissionais de Tecnologia da Informação e é parte importante deste trabalho.

Desenvolver habilidades linguísticas é muito mais do que aprender novos idiomas. É possível que duas pessoas que falem português conversem por alguns minutos sem que haja qualquer compreensão por parte de qualquer das pessoas. A comunicação é mais do que a verbalização de palavras. Para falar um idioma é

necessário apenas conhecer as palavras e estruturas: “A língua dispõe de um certo número de signos fundamentais, arbitrariamente ligados a significações-chave” (MERLEAU-PONTY, 2012, p.30). Conhecer os signos e suas significações-chave permite ao sujeito tentar estabelecer uma comunicação, embora a comunicação efetiva seja bem mais complexa.

Compreensão de um problema

A comunicação começa por compreender o que se diz e por compreender o que é dito pelo outro e o que isso representa para ele. A compreensão do problema do outro pode parecer simples a princípio, mas à medida que a compreensão se aprofunda, surgem novos níveis de complexidade. Poderíamos fazer uma analogia aqui com um jogo de vídeo game, do ponto de vista da percepção do usuário. Nas primeiras fases existem regras simples que bem executadas levam ao avanço para as próximas fases. A execução das tarefas nesta fase inicial parece requerer pouca habilidade para que qualquer iniciante seja capaz de cumpri-las, empenhando apenas um pouco de dedicação e um pequeno esforço repetitivo. A compensação deste esforço é adquirir tamanha habilidade para executar as tarefas simples, que uma vez que elas surgem, o jogador já não precisa mais pensar sobre o que fazer, pois seu cérebro treinado comanda pensamentos e movimentos de forma tão rápida que a tarefa é executada em segundos.

A medida que avança para um nível mais complexo, as tarefas simples, agora executadas automaticamente, tornam-se apenas uma parte das novas tarefas, que devem ser completadas. Por não mais precisar raciocinar sobre as tarefas simples, o jogador passa a executar as habilidades já adquiridas e somar a elas novas ações em pouco tempo. Novamente este processo deve ser repetido até que também essas tarefas sejam internalizadas e automatizadas pelo sujeito deixando-o pronto para um novo nível de complexidade. Este processo se repete e a cada nível aumenta-se o desafio proposto, para que seja cada vez mais difícil automatizar movimentos e raciocínio e sempre que isso ocorre o jogador avança. Se o mesmo jogador volta à fase inicial, sua percepção será a de que as tarefas, antes desafiadoras, tornaram-se monótonas, pois já não impõem qualquer desafio. Por outro lado, se um jogador nos

níveis mais avançados passar seu controle para um jogador iniciante, este, provavelmente não conseguirá avançar e achará impossível que alguém o consiga.

Claro que quando se trata de um jogo, este cenário é projetado para ocorrer desta forma. Entretanto, a nossa aquisição de habilidades muitas vezes ocorre de maneira semelhante, mesmo em ambiente não controlado.

No artigo *Uma Fenomenologia da Aquisição de Habilidades como base para uma Ciência Cognitiva Meleau-Pontyana Não Representacionista*⁷, Hubert L. Dreyfus propõe um modo de compreender este fenômeno. Segundo Dreyfus, existem cinco estágios da aquisição de habilidades: novato, iniciante avançado, competente, proficiente e perito:

- **Novato** - O Novato necessita de regras claras e bem definidas, pois deverá segui-las fielmente a fim de atingir os resultados esperados;
- **Iniciante avançado** - Os iniciantes avançados descobrem que as regras variam de acordo com a situação e que outras regras podem precisar ser aplicadas dependendo das variações que surgirem;
- **Competente** - O Competente começa a questionar as regras que não conseguem determinar o ato a ser efetuado diante de situações mais complexas obrigando o sujeito a tomar decisões com base na sua experiência. Neste ponto existe uma variedade tão grande de situações e particularidades que é impossível prever todas elas e gerar uma regra a ser aplicada a cada situação em função do grande número de variáveis que compõem as situações mais complexas “intérpretes competentes, portanto, devem decidir por si próprios em cada situação que plano ou perspectiva deve adotar sem ter certeza de que ele vai passar a ser apropriado”⁸. A experiência é um fator determinante na escolha das regras a serem aplicadas e a forma de aplicá-las. Nesta fase é muito difícil para o sujeito voltar a seguir regras simples como acontece com sucesso pelo novato;
- **Proficiente** - O proficiente abandona completamente a consulta às regras teóricas na tomada de decisão e passa a resolver os problemas com base nas experiências e no resultado emocional que cada experiência gerou anteriormente. “Apenas se a experiência é assimilada na forma encarnada, não-teórica as reações

intuitivas podem substituir respostas fundamentadas”⁹. É necessário que a resposta seja parte do sujeito, que o corpo dele reaja antes que a mente seja capaz de dizer o que deve ser feito. Assim ocorrem as respostas do proficiente. Ele sabe como proceder sem que tenha que racionalizar o problema ou a solução. Embora ainda precise tomar decisões, e as tome com base em algumas regras, muitas vezes não sabe explicar rapidamente quais regras foram aplicadas a decisão, pois o processo foi altamente intuitivo;

- **Perito** - Ao atingir o nível de perito o sujeito passa a tomar suas decisões de forma totalmente intuitiva. A diferença entre o perito e o proficiente é que o perito consegue distinguir diferenças de plano e perspectiva em situações muito semelhantes. Além disso, o perito, ao se deparar com um problema desconhecido, ainda que não tenha qualquer referência de experiência semelhante em sua memória, consegue propor uma solução, pois quebra imediatamente um problema apresentado em diversos outros problemas menores que sabe como resolver, chegando a uma solução para qualquer problema. Por conseguir fazer esta segmentação de problemas, o perito é o sujeito que desenha as regras para que os novatos possam seguir resolvendo as pequenas partes de um problema muito maior que apenas o perito consegue visualizar em sua forma total para resolver.

Essas fases da aquisição e habilidades impactam diretamente no desempenho dos profissionais de diversas áreas, inclusive profissionais de desenvolvimento de software. Para desenvolver um sistema é necessário buscar a máxima compreensão dos resultados que devem ser obtidos. Pelo que vimos acima, uma equipe composta apenas por peritos conseguiria desenvolver qualquer projeto com perfeição, pois teriam o entendimento máximo do problema e dos resultados a serem obtidos. Entretanto, ao se tratar de um grupo de pessoas, é necessário não apenas compreender o problema, mas certificar-se de que todos os membros da equipe tiveram a mesma compreensão do problema e estejam alinhados em relação à solução a ser desenvolvida. Como cada perito terá seu próprio conhecimento tácito, pode haver um conflito quanto à melhor forma de resolver o problema compreendido, visto que o conhecimento tácito é pessoal e precisa ser transferido.

“a gestão do conhecimento preocupa-se com o conhecimento “tácito”, objetivando o desenvolvimento da capacidade das pessoas em “explicitar” e compartilhar o seu conhecimento” (TARAPANOFF, 1996, p.28).

É necessário considerar ainda, que, se todos os projetos tivessem apenas peritos alocados, além de ter um custo muito alto para desenvolvimento os demais profissionais jamais se tornariam peritos, tornando os peritos ainda mais raros e caros. A solução então deveria se dar de duas formas:

1. Transmissão de conhecimento para os menos experientes;
2. Homogeneização do conhecimento dentro da equipe.

A transmissão do conhecimento tácito, permite que os peritos transmitam sua expertise para os profissionais menos experientes enquanto a homogeneização do conhecimento permite que toda a equipe se alinhe com um objetivo comum.

Transmissão do Conhecimento

Esta questão vem sido altamente debatida nas empresas desde a revolução industrial, quando as tarefas foram segmentadas para que funcionários menos capacitados, e mais baratos, pudessem realizar tarefas mais operacionais, ou que não precisassem de muita tomada de decisão. No caso do desenvolvimento de um sistema, é difícil imaginar que possam existir tarefas que não precisem ser pensadas para serem realizadas. Porém, a falta de mão de obra qualificada e a necessidade de redução de custos fizeram com que mesmo tarefas como o desenvolvimento de um software fossem fragmentadas. Para que profissionais menos experientes pudessem contribuir com o desenvolvimento de tarefas mais simples, foi necessário “decompor a realidade em partes de informação pequenas e simples o suficiente para que a pessoa as processasse” (TAKEUCHI e NONAKA, 2008, p.19). Por exemplo, na equipe de analista de banco de dados existem tanto analista experientes que estruturam e modelam o banco quanto analistas júnior, que podem criar consultas simples ou ainda testar funcionalidades implementadas por analistas mais experientes.

Ainda assim havia necessidade de transmitir a forma de resolver tarefas mais complexas aos aprendizes mais promissores. Funcionários com grande experiência,

grandes peritos, que sabiam resolver os problemas de forma extremamente eficaz e intuitiva, especialmente em áreas técnicas, muitas vezes não conseguiam transmitir seus conhecimentos exatamente por não entender seus processos de solução de problemas ou não saber comunicar seu entendimento, e o mercado passou a exigir cada vez mais essa transferência de conhecimento de forma eficiente “A Habilidade em técnicas de comunicação e apresentações passou a ser competência importante para profissionais de todas as atividades” (BOOG, 2006, p.189).

Visando preencher a lacuna da transmissão de conhecimento, em 1995 surgiu no Japão, uma teoria chamada Gestão de Conhecimento. O objetivo da gestão do conhecimento é transformar o conhecimento tácito em conhecimento explícito. Segundo a teoria o conhecimento explícito “pode ser expresso em palavras, números ou sons, compartilhado na forma de dados, fórmulas científicas...” ou seja, “pode ser rapidamente transmitido aos indivíduos, formal e sistematicamente”. Já o conhecimento tácito “é altamente pessoal e difícil de formalizar, tornando-se de comunicação e compartilhamento difícil” (TAKEUCHI e NONAKA, 2008, p.19).

Os autores indicam ainda que existem quatro tipos de conversão do conhecimento tácito para o conhecimento explícito:

- Socialização (indivíduo para indivíduo) – Compartilhar e criar conhecimento tácito através de experiência direta.

Ocorre quando um perito aceita outro sujeito como aprendiz, ou estagiário, transmite todo o seu conhecimento com demonstrações de como agir diante de cada situação. A vantagem é que o aprendiz aprende e internaliza os processos resolvendo problemas, mas com risco de erro menor em função da presença do perito para auxiliá-lo sempre que um problema ou dúvida surgirem. A desvantagem é que este é um processo demorado e contínuo. Pode levar anos até que o aprendiz se torne um perito. E ainda corre-se o risco de o funcionário sair da empresa durante o processo de aprendizado ou assim que o processo é concluído, levando todo o conhecimento aprendido. É como acontecia a transmissão de conhecimento antes da Revolução Industrial, quando as profissões passavam “de pai para filho”;

- Externalização (indivíduo para grupo) – Articular conhecimento tácito através do diálogo e da reflexão.

Ocorre quando vários indivíduos se juntam e discutem suas ideias e suas percepções sobre um problema de forma aberta a aceitar as ideias dos outros, gerando uma troca de conhecimento onde cada indivíduo contribui com sua experiência e conhecimento gerando a construção de um novo conhecimento criado com base no conhecimento de todos: “Na terminologia da dialética, a síntese de A (tácito) mais B (explícito) cria C (novo conhecimento)” (TAKEUCHI e NONAKA, 2008, p.24). É fundamental nas organizações, visto que apenas pessoas criam conhecimento, portanto o conhecimento da organização é o conhecimento das pessoas que a constituem. O ponto sensível deste modelo é a comunicação que deve acontecer de forma eficiente, sobre a qual falaremos no próximo capítulo;

- Combinação (grupo para organização) – Sistematizar e aplicar o conhecimento explícito e a informação.

É um modelo onde um pequeno grupo de pessoas, normalmente o Arquiteto e o Gerente do Projeto, determinam o conhecimento a ser seguido por todos os outros. Foi o que aconteceu durante a revolução industrial, quando houve a segmentação da informação e a criação de regras e processos. Os processos foram criados e desenhados pela administração das indústrias e impostos aos funcionários. O grande problema deste modelo é que ele corre o risco de não considerar a prática do operador da tarefa, ou, voltando ao desenvolvimento do software, do programador que irá implementar pequenas partes do programa, ou pior, do usuário que irá utilizar o sistema. Não há troca de informação e há pouco conhecimento gerado;

- Internalização (organização para indivíduo) – Aprender e adquirir novo conhecimento tácito na prática

É o processo empírico de aprendizagem. O sujeito tenta sozinho resolver problemas, verifica a eficiência da tentativa, e tenta novamente, alterando o processo caso este não tenha funcionado. Embora o aprendizado fique enraizado no indivíduo e gere uma grande quantidade de conhecimento sobre como fazer algo e também sobre o que não deve ser feito, o tempo de aprendizagem é muito

grande e impraticável no mundo corporativo considerando que os erros possuem um alto custo para as organizações.

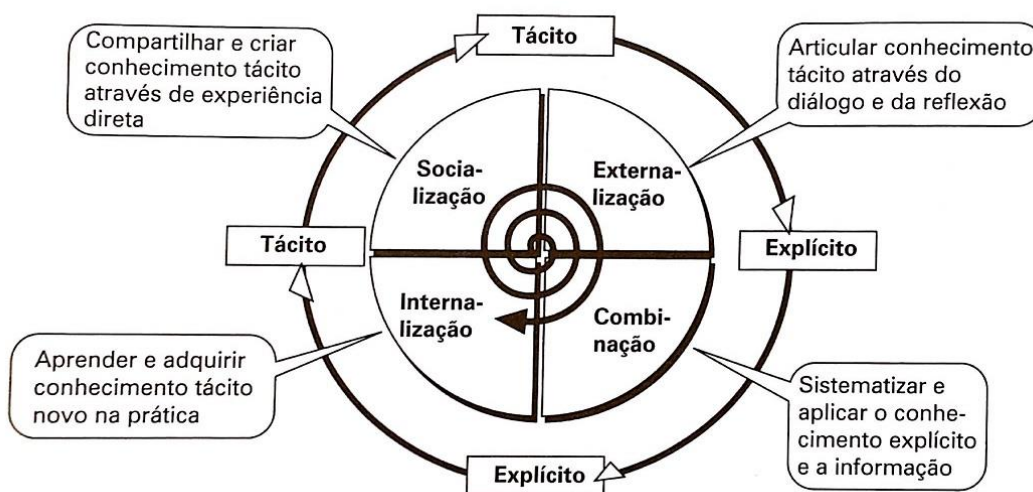


Figura 7 - Processo SECI. Fonte: Adaptada de Nonaka e Takeuchi, 1995.

Sem dúvida alguma, no processo de desenvolvimento de software o modelo mais adequado para a compreensão do problema proposto e implementação de soluções é o da Externalização. Reunir todos os participantes de um projeto para efetuar o conhecido *brain-storm* proporcionando a maior troca de conhecimento possível entre os indivíduos. Nesses encontros ocorrem elucubrações, muitas vezes de assuntos que não estão diretamente relacionados ao tema, mas de que de alguma forma contribuem para a construção do pensamento. O ponto nevrálgico e objeto deste estudo é como estabelecer uma comunicação efetiva entre pessoas que essencialmente são lógicas e racionais, treinadas para recortar um problema e tratá-lo com a maior objetividade possível, deixando de lado o pensamento abstrato ou a percepção do outro que não esteja modulado e formalizado.

Há décadas se fala em habilidades dos profissionais de desenvolvimento de software sem mencionar suas habilidades em comunicação. O livro *Downsizing em sistemas de informação: Reengenharia de sistemas de informação* (GUENGERICH, 1993) é um bom exemplo a ser citado. No livro, Guengerich explica detalhadamente

todos os passos para a migração de mainframes para microcomputadores. Fala sobre os passos para o desenvolvimento e manutenção de softwares, além de detalhar custos, hardware, gerenciamento de redes e outros temas fundamentais a computação do início da década de noventa.

Em um dos capítulos chamado “Montando uma equipe para desenvolvimento de aplicativos distribuídos” são apresentados os pré-requisitos que cada profissional da equipe deve apresentar. O livro coloca o programador como a figura central da equipe, “O desenvolvedor de aplicativos distribuídos é responsável pelo esforço central do processo de desenvolvimento” (GUENGERICH, 1993, p. 134). São apresentadas também as funções do gerente de projetos, “O Gerente do projeto é responsável pela supervisão da equipe de projeto, pelo planejamento e distribuição de trabalho e pela garantia de qualidade”.

Em “Habilidades da equipe de trabalho”, Guengerich lista “Um excelente conhecimento do projeto de dados relacionais...” e “Compreensão conceitual das redes...” (GUENGERICH, 1993, p. 136) e detalha ainda, quatro níveis de habilidades essenciais aos profissionais da equipe a serem tabulados em uma matriz: *nível conceitual, trabalha sob supervisão, trabalha independente com a tecnologia e especialista verdadeiro*.

Guengerich reconhece a importância da comunicação para o sucesso do projeto. Ele entende que é impossível uma única pessoa possuir todas as habilidades necessárias ao desenvolvimento de um software: “O desenvolvimento de todas essas técnicas em uma só pessoa seria um processo proibitivamente demorado e caro” (GUENGERICH, 1993, p. 133). E define os passos para resolver a questão: “você incorpora tais conhecimentos técnicos entre os membros da equipe, providencia para que haja boas comunicações entre os membros da equipe e faz uso de uma metodologia confiável” (GUENGERICH, 1993, p. 133). O ponto a ser observado aqui é que “providencia para que haja boas comunicações” além de ser a única orientação neste sentido, é muito vago.

As funções executadas por cada pessoa da equipe de desenvolvimento pouco foram alteradas nas últimas duas décadas. Ainda existem, ou deveriam existir em um projeto o gerente do projeto, o analista de banco de dados, o programador, a pessoa responsável pelos testes e o web designer entre outros. Entretanto, permanecem as

dificuldades na condução e finalização dos projetos desenvolvidos por pequenas e médias empresas. Os problemas existem nas grandes empresas também, mas não são o objeto deste estudo. Nossa questão é tentar entender porque esses problemas ainda existem, se temos tanto conhecimento a respeito do que deve ser feito. Se sabemos que o conhecimento deve ser transmitido para as outras pessoas e temos o conhecimento, porque ainda há tantos problemas na entrega dos projetos?

Precisamos olhar para a forma como nos comunicamos. Talvez um caminho seja tentar entender como a comunicação se dá, para que possamos analisar se encontramos o caminho correto para melhorarmos as relações cliente-equipe e entre as pessoas de uma equipe envolvida no desenvolvimento de software em uma pequena ou média empresa.

COMUNICAÇÃO

Ao analisar as relações entre pequenas e médias empresas e os meandros do desenvolvimento de software, observamos que existe uma deficiência nas relações entre pessoas de uma equipe e entre clientes e fornecedores. Essa relação, que chamaremos aqui de *diálogo* é vital para o perfeito entendimento entre dois profissionais.

Neste capítulo vamos olhar como a filosofia compreende a questão do diálogo. Através do olhar de dois filósofos importantes que se debruçaram sobre o tema, Maurice Merleau-Ponty e Martin Buber, vamos analisar como o diálogo se estabelece.

Nos processos de desenvolvimento de software, como vimos anteriormente o foco é implementar um código sem erros, baseado no melhor entendimento dos objetivos a fim de atender a uma demanda do cliente que muitas vezes não está clara nem para ele próprio. Segundo o fenomenólogo francês Merleau-Ponty, no livro *A Prosa do Mundo*, o algoritmo e a ciência exata tratam das coisas sem considerar o repertório que os interlocutores possuem para interpretá-las. E ainda assim o processo de aquisição de conhecimento só acontece quando o interlocutor utiliza seu repertório para criar significado naquilo que lhe foi apresentado, mesmo se tratando de ciências exatas, como a matemática: “O sentido novo só sai do sentido antigo por uma transformação que se dá fora do algoritmo” (MERLEAU-PONTY, 2012, p.215).

O primeiro passo para se estabelecer um diálogo seria, então, compreender o sentido do que está sendo dito. Do mesmo modo, no desenvolvimento do software deve ser compreendido o sentido do que está sendo criado. Quando criamos significados para as coisas e palavras, elas se transformam. Merleau-Ponty diz que o sentido que surge através de um texto ou de um algoritmo desenvolvido por outra pessoa apenas será compreendido quando olharmos essa significação como algo que não surge nem no algoritmo ou no texto e nem no interlocutor, precisamos compreender como algo que surge de uma intersecção dos conjuntos de conhecimento: “Só compreenderemos inteiramente esse salto sobre as coisas em direção ao seu sentido, se o compreendemos como invasão de mim sobre o outro e do outro sobre mim...” (MERLEAU-PONTY, 2012, p.218). A troca que se dá *entre* os sujeitos cria sentido. O diálogo proporciona esse salto de um sobre o outro. Possibilita

a troca que irá por fim gerar novos significados. Entretanto, quando *um* fala, não sabe se o outro está ali e se compreende o que, de fato, ele quer dizer. As palavras são ditas ao vento diante de outra pessoa e não sabemos se as compreende e sequer se as escuta. Apenas quando percebemos alguns gestos, sons e murmúrios temos, mesmo sem palavras, a sensação de que fomos ouvidos, de que fomos compreendidos: “Os olhares que eu lançava pelo mundo como o cego tateia os objetos com seu bastão, alguém os pegou pela outra ponta e os retorna contra mim, para, por sua vez, tocar-me” (MERLEAU-PONTY, 2012, p.220). Para que haja o diálogo é necessário que possamos perceber que o outro nos notou, nos percebeu.

Para que essa comunicação ocorra é necessário observar o outro. Ouvir o outro. Entender o ponto de vista do outro sem julgar ou criticar. Precisamos entender, do mesmo modo como cada um, à sua maneira, traz uma contribuição para o entendimento do mundo, o mesmo se dá quando do desenvolvimento de um projeto. E, ao observar o outro, percebemos a nossa verdade de uma forma diferente. Para Merleau-Ponty nós falamos sempre para outra pessoa. Nos dirigimos ao outro na esperança de que ele nos compreenda. Intimamente, desejamos e esperamos que ele possa nos compreender porque ele é igual a nós. Um ser feito da mesma matéria de que somos feitos, que vive no mesmo mundo em que vivemos e por isso deveria entender o que dizemos, pois o que dizemos é aquilo que pensamos e que o outro deveria compreender, já que ele pensa também. No entanto, não estamos certos de que o outro realmente capte nossos pensamentos, então passamos a observar o nosso observador na esperança também de compreender se ele nos compreendeu. Dessa forma, olhamos o outro como antes olhávamos a nós mesmos “É no mais íntimo de mim que se produz a estranha articulação do outro” (MERLEAU-PONTY, 2012, p.221).

Para Merleau-Ponty cada pessoa possui o que ele chama de “campo”: “Não há lugar para ele senão em meu *campo*, mas esse lugar, pelo menos, está preparado para ele desde que comecei a perceber” (MERLEAU-PONTY, 2012, p.223). O *campo* é algo que extrapola o nosso corpo. É todo o nosso conhecimento, nossa experiência e percepção da vida. A cada nova experiência inserimos e modificamos o nosso *campo*, que impacta diretamente em quem somos e, dessa forma, modificamos a nós mesmos. A interação com outras pessoas através do diálogo modifica o nosso *campo* à medida em que recebemos parte do *campo* de outros no nosso próprio *campo* e

assim, nos modificamos junto com o outro, criando um conhecimento que não seria possível a nenhum dos dois sozinho, pois aquele conhecimento que se forma não pertencia a nenhum dos dois. A comunicação então, ocorre neste momento em que o *campo* de um e do outro se tocam e se atravessam.

Nós nos projetamos no mundo, no outro e em tudo que no mundo existe buscando a nossa verdade nesse mundo comum contido no *campo* de cada pessoa.

A fala é o instrumento que utilizamos para extravasar os limites do nosso *campo* e adentrar o mundo do outro com nossos pensamentos, assim como é a forma como o outro adentra ao nosso mundo “...nossa fala vai, como o atestam as respostas, tocar nele suas significações, invadindo-nos um ao outro na medida em que pertencemos ao mesmo mundo cultural...” (MERLEAU-PONTY, 2012, p.228). O idioma é a forma comum de darmos nome as coisas para que a troca de palavras facilite a geração de significados. Não que o significado esteja nas palavras, mas as palavras passam a ser capazes de representar ou de buscar no outro o significado ali presente, visto que representam uma imagem comum a ambos: “Cumpra portanto conceber sua operação fora de toda significação já instituída, como o ato único pelo qual o homem que fala se atribui um ouvinte, e uma cultura que lhes seja comum” (MERLEAU-PONTY, 2012, p.228). E quando a comunicação funciona, as palavras ganham sentido e o significado se estabelece no *campo* de cada indivíduo.

Já para o filósofo judeu alemão Martin Buber, o homem existe a partir da fala. É a fala que faz com que ele perceba sua existência “Quando o homem fala, quando profere palavras, quando enuncia, o ser se atualiza nele; através dela, ele “faz-se homem” e situa-se no mundo com os demais homens”.

Buber fala do homem *dialógico* que acolhe sua função de interagir com os outros e com o universo que o cerca para que possa de fato *ser*:

“Mesmo que se possa prescindir da fala, da comunicação, há, contudo, um elemento que parece pertencer indissolavelmente à constituição mínima do dialógico, de acordo com seu próprio sentido: a reciprocidade da ação interior” (BUBER, 2006, p.42).

Segundo Buber, palavras não são necessárias para que ocorra o diálogo. O diálogo já ocorre quando há uma percepção de uma situação que nos toca e nos transmite alguma informação que até então não nos pertencia. A simples presença do outro quando o observamos pode nos *dizer* algo:

“As coisas acontecem de outra maneira quando, numa hora receptiva da minha vida pessoal, encontro-me um homem em quem há alguma coisa, que eu nem consigo captar de uma forma objetiva, que “diz algo” a mim” (BUBER, 2006, p.42).

Mas apenas palavras também não bastam para que ocorra o diálogo, pois palavras soltas que não nos *tocam* e não geram para nós algum significado não cumprem o papel do diálogo. São como a grama onde pisamos sem sequer perceber que existe. Não muda nada em nossa existência, até que um dia olhamos para ela, sentimos sua textura e, de repente, aquela grama que sempre esteve ali passa a existir para nós. Assim são as palavras que não detectamos “Signos nos acontecem sem cessar. Viver significa ser alvo da palavra dirigida; nós só precisaríamos tornar-nos presentes, só precisaríamos perceber” (BUBER, 2006, p.43). Para perceber as palavras que nos são dirigidas precisamos nos conectar nessa frequência onde os receptores captam o significado das palavras.

“Os signos da palavra dirigida a alguém não são algo de extraordinário, algo que se destaca da ordem comum das coisas; são justamente o que se passa de tempo em tempo, justamente o que se passa de qualquer maneira, nada lhes é acrescentado pela palavra dirigida” (BUBER, 2006, p.43).

Nem mesmo estar entre os homens é suficiente, pois o homem pode estar entre outros e não sair do seu próprio universo “Não é a vida do homem solitário que devemos chamar de monológica, mas daquele que não é capaz de atualizar, de uma forma essencial, a sociedade na qual o seu destino o faz mover-se” (BUBER, 2006, p.54).

Mesmo estando sozinho o homem pode se conectar à natureza, aos outros homens que estão à sua volta e a todo o ambiente que o cerca observando e estando receptivo às influências externas e à forma como essas influências podem fazê-lo perceber sua própria existência, sua própria condição e, dessa forma, transformá-lo em alguém diferente:

“A existência dialógica recebe, mesmo no extremo abandono, uma sensação áspera e revigorante de reciprocidade; a existência monológica não se aventurará, mesmo na mais terna comunhão, a tatear para fora dos contornos de si mesma” (BUBER, 2006, p.55).

Segundo Buber, todas as coisas se tornam pessoais para nós a medida que as colocamos em perspectiva. E ao observá-las, estabelecemos uma relação, pois a nossa observação é sempre no sentido de nos levar ao encontro do que é observado. No magnífico texto abaixo, extraído do livro *Eu e Tu*, Buber demonstra essa percepção:

“Eu considero uma árvore.

Posso aprendê-la como uma imagem. Coluna rígida sob o impacto da luz, ou o verdor resplandecente repleto de suavidade pelo azul prateado que lhe serve de fundo.

Posso senti-la como movimento: filamento fluente de vasos unidos a um núcleo palpitante, sucção de raízes, respiração das folhas, permuta incessante de terra e ar, e mesmo o próprio desenvolvimento obscuro.

Eu posso classificá-la numa espécie e observá-la como exemplar de um tipo de estrutura e de vida.

Eu posso dominar tão radicalmente sua presença e sua forma que não reconheço mais nela senão a expressão de uma lei – de leis segundo as quais um contínuo conflito de forças é sempre solucionado ou de leis que regem a composição e a decomposição das substâncias.

Eu posso volatilizá-la e eternizá-la, tornando-a um número, uma mera relação numérica.

A árvore permanece, em todas estas perspectivas, o meu objetivo tem seu espaço e seu tempo, mantém sua natureza e sua composição.

Entretanto pode acontecer que simultaneamente, por vontade própria e por uma graça, ao observar a árvore, eu seja levado a entrar em relação com ela; ela já não é mais um Isso. A força de sua exclusividade apoderou-se de mim.”

(Martin Buber, 2006, p.54)

É importante esclarecer que, quando se trata do diálogo, não se trata de uma concordância. A Comunicação não surge apenas quando dois indivíduos entregam seus pensamentos e percepções ao mundo, e, de forma harmoniosa, surge uma nova percepção para ambos. Muitas vezes, a discordância leva a um aprendizado ainda maior. Quando temos o nosso ponto de vista questionado e confrontado, abre-se uma janela à nossa frente e uma luz ilumina caminhos que não conhecíamos ou aos quais não tínhamos intenção de percorrer “[...]mesmo a objeção que o interlocutor me faz me arranca pensamentos que eu não sabia possuir” (Merleau-Ponty, 2011, p.475).

Em uma equipe, as famosas reuniões de *brain-storm* podem exaltar ânimos quando surgem discordâncias, pois somos apegados aos nossos pontos de vista. Entretanto, essas reuniões, quando bem mediadas e conduzidas, podem produzir uma grande quantidade de conhecimento para todos os profissionais. Neste “diálogo” entre muitas pessoas, aprendem não só os que se entregam à fala e à discussão, mas também aqueles que observam os que se comunicam e que mesmo em silêncio se transformam.

Todos nós temos a nossa própria visão do mundo, e em geral, ela parece bastante correta, pois quanto mais nos fechamos dentro dela, mais analisamos e encontramos (ou criamos) lógica e coerência para o nosso modo de pensar e agir. Criamos justificativas para explicar o que fazemos e a forma como fazemos. Assim nos mantemos dentro da nossa zona de conforto. Apenas quando olhamos o outro, temos nossas verdades confrontadas. Aquilo em que acreditávamos passa a ficar explícito que não era a melhor maneira de se fazer alguma coisa. E com a realidade explícita diante de nós, vemos um argumento contra as nossas justificativas e assim, nos obrigamos a questionar e, por vezes, mudar nosso ponto de vista criando uma nova realidade a ser confrontada futuramente ao olharmos para outro *outro*.

A comunicação permite essa troca, esse olhar que nos faz aprender e também contribuir com o outro. Porém, se resolvermos o problema da comunicação, e apurarmos nosso olhar e nossa percepção sem medo de confrontar aquilo em que acreditamos, e que é mais confortável para nós, ainda temos a necessidade de explicitar os nossos pensamentos. Como dito antes, em geral, os profissionais de TI possuem suas habilidades lógico-matemáticas mais desenvolvidas do que as habilidades linguísticas.

Visando facilitar o processo de explicitação de conhecimento diversas técnicas foram desenvolvidas e deveriam ser utilizadas por todos que precisam compartilhar o entendimento de um problema, técnico ou não. No caso do desenvolvimento de software, onde existem uma série de regras a serem respeitadas e sequências a serem seguidas, foram criadas técnicas de modelagem de processos que podem ser de grande auxílio na comunicação entre todas as pessoas da equipe.

No próximo capítulo vamos entender como funcionam duas dessas técnicas.

MODELAGEM DE PROCESSOS

Pequenas e médias empresas têm a necessidade de otimizar seus recursos para serem mais competitivas e mesmo para sobreviverem às dificuldades de atuação no mercado brasileiro.

“Uma empresa de software bem-sucedida é aquela que fornece software de qualidade e capaz de atender às necessidades dos respectivos usuários. Uma empresa que consiga desenvolver este software de maneira previsível e em determinado período, com utilização eficiente e eficaz de recursos, será uma empresa com um negócio viável” (BOOCH, 2012, p.3).

Sem condições de oferecer aos seus colaboradores os mesmos salários e benefícios que as grandes empresas oferecem, as pequenas e médias empresas buscam opções para se tornarem atrativas também para profissionais qualificados. Entre as vantagens oferecidas aos colaboradores, estão a possibilidade de crescimento junto com a empresa, aumento de salários por meritocracia e participação na tomada de decisão. Além disso, as empresas menores, em geral, oferecem aos colaboradores uma autogestão de tempo com cobranças baseadas mais nas entregas do que no tempo de permanência no escritório. O home office, por exemplo é uma tendência crescente, que em função da falta de regulamentação específica dificulta sua aplicação em grandes empresas que são sujeitas a auditorias constantes e regras de *compliance* mais rígidas.

Diante deste cenário, algumas empresas menores começaram a buscar formas de melhorar o desenvolvimento de sistemas, para ajudar a otimizar o tempo dos colaboradores, gerando vantagens para a empresa que fará suas entregas em menor tempo e com mais qualidade; e também para o profissional, que enfrentará uma pressão menor em relação ao prazo de entrega, e consequentemente terá condições de fazer seu trabalho com tranquilidade, contribuindo também para o aumento da sua qualidade de vida.

O desenvolvimento de um software, assim como a criação de qualquer outra coisa, exige algum planejamento. Muitas vezes, este planejamento ocorre apenas no nível mental. Para preparar um alimento trivial para os brasileiros, como arroz, por

exemplo, é necessário pegar todos os ingredientes e utensílios necessários à preparação. Mesmo que o cozinheiro tenha a receita na cabeça e não precise consultá-la, ou mesmo que nunca a tenha lido, que tenha aprendido com alguém mais experiente, por alguns segundos irá repassar o preparo em sua cabeça e verificar se possui tudo o que é preciso para preparar o arroz. Em seguida irá iniciar o processo em uma ordem específica, sempre verificando e fazendo ajustes necessários de temperatura, quantidade de água, temperos, etc.

Quanto mais simples for a tarefa, mais fácil é fazer o planejamento no nível mental e transmitir aos outros a metodologia utilizada. Seguindo o nosso exemplo, se o mesmo cozinheiro, decidir um dia fazer um risoto pela primeira vez ele vai se deparar com uma realidade bem diferente daquela a que está acostumado. Se não tiver uma orientação (receita) com instruções detalhadas de como prosseguir, com ingredientes, sequência e resultado a ser esperado a cada etapa de preparação, será bastante difícil alcançar um resultado satisfatório. O cenário fica pior quando o cozinheiro começa a preparar o arroz no modo tradicional, e após iniciar a preparação descobre que seu cliente, na verdade queria um risoto. Nesse momento, o cozinheiro se vê em uma situação bem difícil, onde tem que escolher entre interromper o que estava fazendo, perder os ingredientes que havia utilizado, lavar os utensílios e iniciar a preparação do risoto do zero, ou tentar transformar o arroz convencional em risoto.

Para que este tipo de problema não ocorra, é necessário ter um bom entendimento da expectativa do cliente, conforme dito no capítulo 2, e uma metodologia que ajude no planejamento e estruturação do projeto como um todo; além de um processo de desenvolvimento que permita alguns ajustes de direcionamento mesmo após o início do desenvolvimento, quando necessário, como afirma Booch.

“Para desenvolver software de qualidade duradoura, será necessário criar uma arquitetura de fundação sólida que aceite modificações” (BOOCH, 2012, p.3).

Uma parte muito importante do processo de desenvolvimento de sistemas é a modelagem. Através da modelagem é possível mapear e documentar todas as etapas do processo de desenvolvimento, comunicando a estrutura e o comportamento desejados pelo sistema a todos os envolvidos no processo, facilitando a comunicação e consequentemente o entendimento do projeto como um todo, e ainda, de cada item que compõe o todo. Os sistemas podem ser desenvolvidos por um grande número de

pessoas e podem ser compostos por um grande número de etapas e módulos distintos. Modelar cada um desses módulos nos ajuda a ter o entendimento de cada detalhe que precisa ser entregue para que o sistema esteja completo.

Escrever um bom software não significa escrever muitas linhas de código. Pelo contrário, significa escrever apenas as linhas de código necessárias. Um projeto mal estruturado, muitas vezes resulta em linhas de código desnecessárias que só dificultam o entendimento dos programadores no momento da manutenção, além de impactar no tempo e aumentar o risco de erros.

A modelagem é altamente utilizada em diversas áreas como arquitetura, marketing, hardware e até na moda que é um processo artístico. Um estilista faz uma prancha conceitual onde coloca elementos que utilizou como inspiração, faz desenhos dos resultados possíveis, esboça o desenho final de uma peça e costura uma peça, ou um modelo, para verificar como será o produto final. Neste caso, não há exatamente uma receita a ser seguida ou sequer um resultado final bem desenhado: o resultado muitas vezes, acontece ao longo do processo. Quando se trata do desenvolvimento de sistemas, por outro lado, o resultado esperado já é sabido, ou deveria ser. O que precisamos é apenas detalhar todos os pontos para planejar cada entrega: “O modelo é uma simplificação da realidade” (BOOCH, 2012, p.5).

Quando se trata de modelagem de software, dois aspectos principais devem ser considerados no planejamento: a interface que irá permitir a interação com o sistema projetado, e o comportamento que o sistema deve apresentar. Para ambos, o entendimento do cliente é muito importante. Entretanto, quando se trata do desenvolvimento de um software sob demanda, especialmente se desenvolvido por uma pequena ou média empresa, a interface deve ser planejada para atender e agradar ao máximo o cliente, permitindo, se possível, customizações - de forma que, no caso de uma eventual troca de interlocutor no cliente, o sistema possa ser configurado para agradar ao novo usuário.

Os modelos de interface devem conter então desenhos das telas finais e protótipos como wireframe, no caso de websites, para alinhamento da expectativa do cliente e também para facilitar o alinhamento e a comunicação entre as pessoas da equipe sobre aquilo que deve ser realizado, evitando interpretações pessoais e errôneas a respeito do projeto. Sem esse direcionamento o programador pode

estruturar seu código de uma forma que não é condizente com a forma com que o analista de banco de dados pensou em estruturar suas entradas e saídas de dados para a mesma tela, e o web designer pode desenhar uma tela que não condiz com nenhum dos dois, por exemplo.

A modelagem ajuda a aumentar o nosso nível de abstração, facilita o planejamento do sistema, ajuda a visualizar o que será entregue, e a entender o comportamento que o sistema deve apresentar. Ela guia a equipe no desenvolvimento e permite documentar as decisões tomadas ao longo do desenvolvimento do sistema. A formalização da modelagem gera histórico do projeto para alterações futuras e também para o desenvolvimento de outros projetos, pois nos ajuda a desenhar um processo de desenvolvimento.

Aqueles processos mais simples cujo planejamento ocorre no nível mental como a preparação do arroz utilizada como exemplo no início deste capítulo, nos deixa dependentes do especialista que está executando a tarefa. Isso significa que cada especialista irá efetuar as coisas do seu jeito, gerando não só uma dificuldade quando for necessário efetuar uma ampliação ou alteração do sistema, mas também, perdendo o histórico do que foi feito e porque foi feito desta forma, no caso de o profissional não estar mais disponível no momento em que a demanda surgir. Para evitar esses problemas os especialistas precisam comunicar e transferir seus conhecimentos para outras pessoas.

A modelagem de processos é uma técnica que permite a uma pessoa detalhar seu conhecimento para que este possa ser comunicado, transmitido para outras pessoas – sejam aprendizes, ou pessoas do mesmo nível que precisam ter conhecimento do projeto. Além disso, a modelagem do processo faz com que quem a desenha, observe atentamente a forma como faz as coisas o que permite questionar e ajustar o próprio processo.

Existem diversas metodologias de modelagem. Este trabalho busca entender se seriam os processos de modelagem a resposta para melhorar a comunicação entre os integrantes de uma equipe de desenvolvimento de sistemas de pequenas e médias empresas. Técnicas formais de modelagem são, muitas vezes, negligenciadas por empresas deste porte por falta de estrutura ou de organização. O resultado é um software mais artesanal e com obtenção de certo sucesso, talvez por ter menos

pessoas envolvidas no processo. Entretanto, ocorre uma perda muito grande no aproveitamento de software, em função da falta de um arquiteto de software ou de metodologias corretas para modelagem dos sistemas. Como este trabalho não trata de técnicas de programação, mas sim de como as pessoas envolvidas no desenvolvimento podem chegar a um melhor entendimento do escopo do sistema e de suas entregas de acordo com a expectativa do cliente, não vamos entrar no mérito da modelagem da forma de desenvolvimento, mas sim, do comportamento do sistema como necessidade de entrega de um resultado específico ao cliente. Para tanto iremos estudar duas formas de modelagem UML (Unified Modeling Language) e BPMN (Business Process Modeling Notation).

UML

UML foi desenvolvida com o objetivo de se criar uma linguagem única para que todos os desenvolvedores pudessem compreender códigos criados por outros profissionais. Além disso, a linguagem foi desenvolvida para possibilitar o entendimento e a comunicação das regras de negócio e das atividades a serem implementadas no projeto, “Ela poderá ser empregada para visualização, a especificação, a construção e a documentação de artefatos que façam uso de sistemas complexos de software” (BOOCH, 2012, p.14). Poderá ser utilizada para modelagem de diversos tipos de sistema e abrange todas as visões necessárias ao desenvolvimento e implantação de sistemas.

É controlada pelo OMG (Object Management Group), um consórcio aberto de empresas formado para estabelecer padrões que suportassem interoperabilidade, especificamente a de sistemas orientados a objetos.

Cada pessoa pode utilizar UML com um objetivo diferente. Fowler, entretanto, detalhou as características de três usos para a UML 2: *esboço*, *projeto* e *linguagem de programação*. Sendo o esboço, na visão de Fowler o mais comum. “Nessa utilização, os desenvolvedores usam a UML para ajudar a transmitir alguns aspectos de um sistema” (FOWLER, 2005, p. 26). Segundo FOWLER, é possível utilizar esboços no desenvolvimento e na engenharia reversa. No desenvolvimento, desenha-se um diagrama UML antes de escrever o código, enquanto a engenharia reversa constrói um diagrama UML a partir de um código já existente, para ajudar seu entendimento.

Segundo BOOCH, UML é uma linguagem destinada a:

- Visualizar;
- Especificar;
- Construir;
- Documentar os artefatos de um sistema complexo de software.

As linguagens gráficas de modelagem existem a bastante tempo com o objetivo de facilitar o entendimento do projeto, já que as linguagens de programação não

possuem nível de abstração alto o suficiente para isso. A UML permite que seja criada uma representação física e conceitual de um sistema, compreendendo seus itens e segmentando-os em submodelos que podem ser agrupados em pacotes de modelos.

“Um processo bem definido servirá como guia para decidir quais artefatos serão produzidos, quais atividades e trabalhadores serão escolhidos para criá-los e gerenciá-los e como esses artefatos serão empregados para medir e controlar o projeto como um todo”. (BOOCH, 2012, p.15)

Entretanto a UML não é uma linguagem de programação e sim uma linguagem para construção de modelos que podem ser conectados a várias linguagens de programação.

A questão da comunicação, objeto de estudo deste trabalho, é citada em diversas bibliografias que descrevem a UML. A modelagem tem como objetivo ajudar os desenvolvedores a entrar em um consenso sobre todas as partes de códigos que devem ser desenvolvidos para atingir o objetivo final. A UML pode ser utilizada tanto para a modelagem do software quanto para a modelagem conceitual: quando utilizada para a modelagem de software, serve de guia para o programador escrever o código fonte de cada item modelado; quando utilizada para a modelagem conceitual, facilita o entendimento geral do projeto e também de cada item modelado possibilitando um melhor entendimento e troca de informações entre as pessoas da equipe.

“No esboço para desenvolvimento, você delineaia alguns problemas em código que você está prestes a escrever, normalmente discutindo-os com um grupo de pessoas da sua equipe. Seu objetivo é usar os esboços para ajudar a transmitir as ideias e alternativas sobre o que está prestes a fazer”. (FOWLER, 2005, p. 26)

Documentação através de UML

A documentação de um sistema é fundamental para a manutenção deste e também para o reaproveitamento do sistema existente para desenvolvimento de um novo sistema ou mesmo de novas funcionalidades dentro do mesmo sistema. Todo sistema deveria ter:

- Requisitos;
- Arquitetura;
- Projeto;
- Código-fonte;
- Planos do Projeto;
- Testes;
- Protótipos;
- Versões.

Nem todos esses passos são formalizados e documentados adequadamente. Utilizando UML é possível documentar todos esses passos, armazenando o histórico de forma organizada e fácil de armazenar e localizar.

Diagramas

A modelagem do comportamento de um sistema nos ajuda a explicitar as regras de negócios que o sistema deve executar, estruturando as interações entre os objetos do sistema. Os *diagramas* nos ajudam a especificar os aspectos dinâmicos dos sistemas.

Tanto na modelagem de software quanto na modelagem conceitual são utilizados diagramas. Segundo Fowler a maioria dos usuários da UML, particularmente os profissionais que fazem esboços, vê a essência da UML como sendo os diagramas, entretanto, “Os criadores da UML veem os diagramas como secundários; a essência da UML é o metamodelo”. (FOWLER, 2005, p. 29) A figura 1.1 é uma pequena parte do metamodelo UML que mostra o relacionamento entre características.

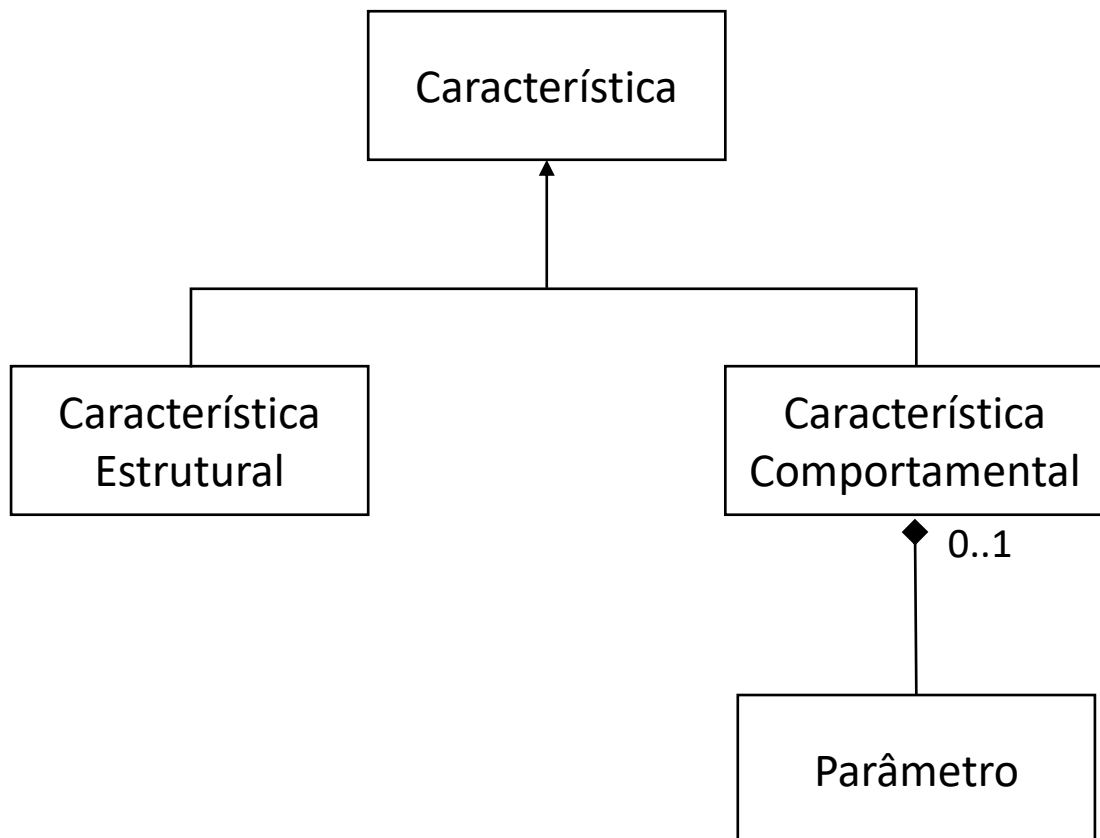


Figura 8 - Uma pequena parte do metamodelo UML. Fonte: FOWLER (2005, p. 32)

Uma das maiores mudanças que a UML 2.0 trouxe em relação às anteriores, além de mudanças profundas no metamodelo, foi a inclusão de novos tipos de diagramas. A estrutura final de diagramas ficou assim:

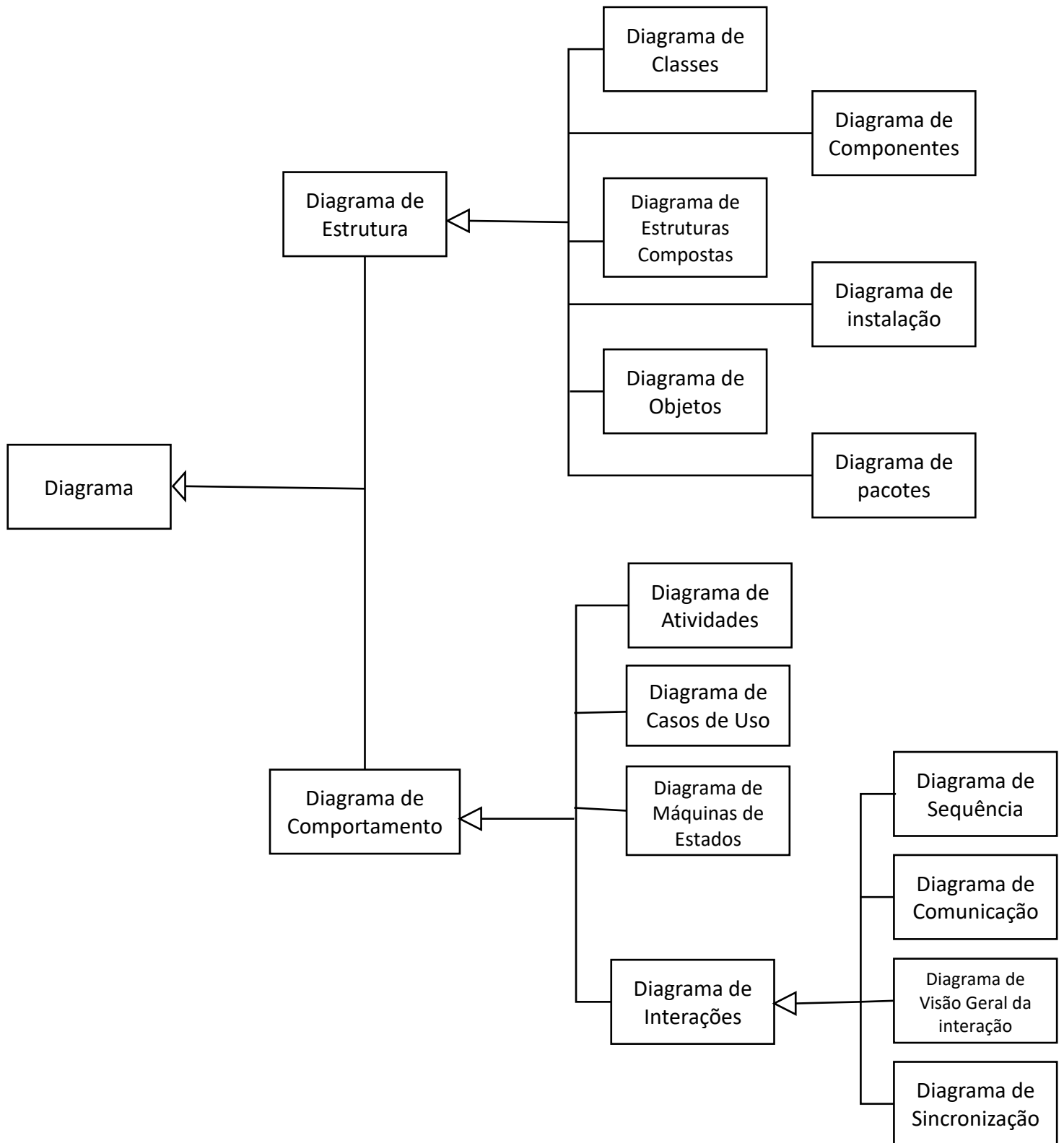


Figura 9 - Classificação dos tipos de Diagrama da UML. Fonte: FOWLER (2005, p. 148)

Diagramas de Comportamento

Como objetivo deste trabalho é estudar a comunicação e o entendimento do comportamento geral do sistema por todas as pessoas envolvidas no desenvolvimento de um sistema, iremos detalhar aqui apenas os diagramas de comportamento.

Diagrama de Atividades

O diagrama de atividades determina a ordem essencial de sequência que se deve seguir. Ele deve descrever a lógica de procedimento de cada atividade, processo de negócio e fluxo de trabalho. Através deste diagrama é possível descrever se existe dependência entre atividades ou se elas podem ser executadas em paralelo. “O diagrama de atividades permite que quem está seguindo o processo escolha a ordem na qual fazer as coisas” (FOWLER, 2005, p. 118).

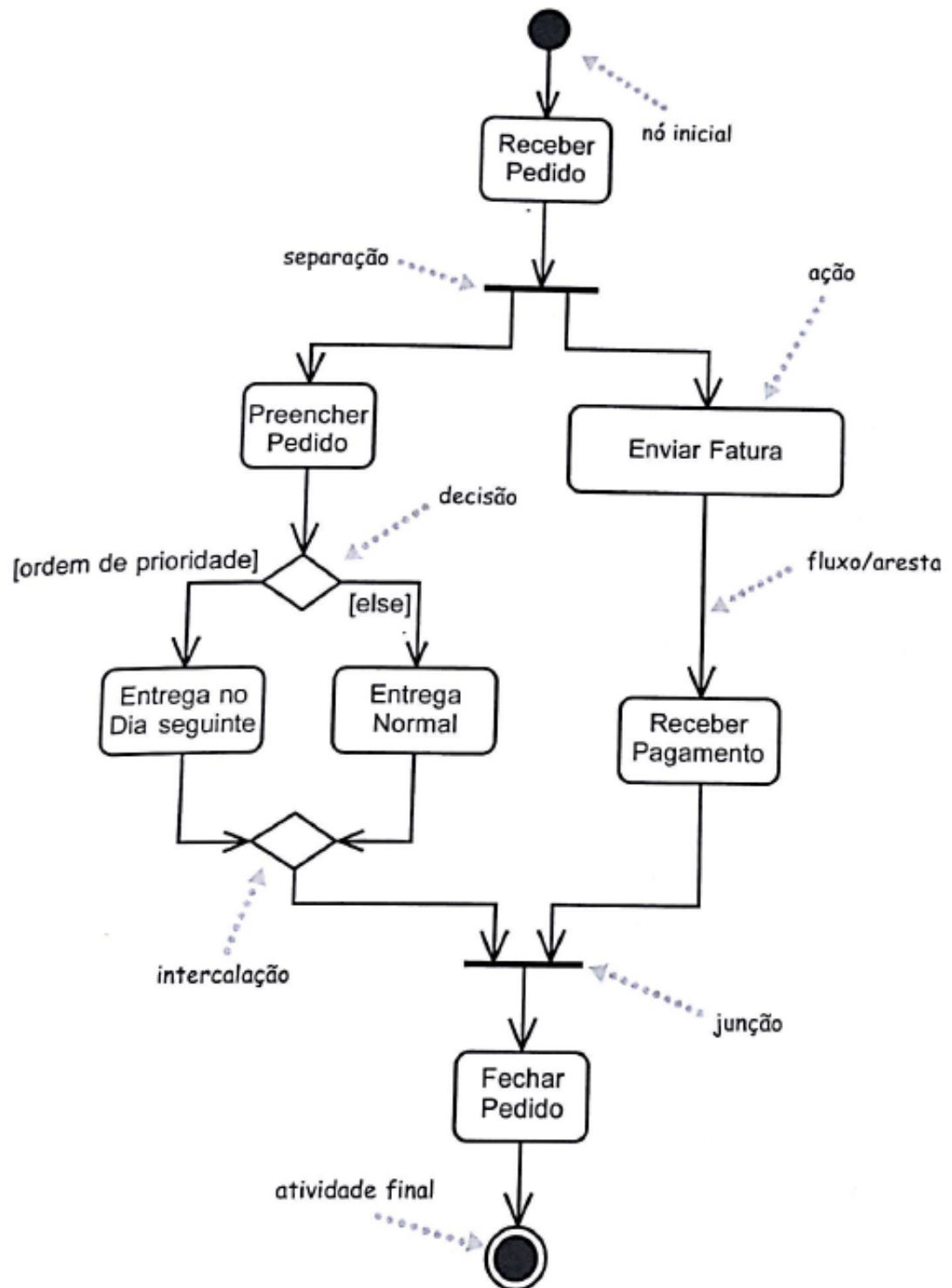


Figura 10 - Diagrama de Atividade Simples. Fonte: FOWLER (2005, p. 119)

Diagrama de Casos de Uso

Os diagramas de casos de uso são utilizados para descrever as funcionalidades de um sistema. Devem detalhar as interações do usuário com o sistema especificando os cenários possíveis “Um cenário é uma sequência de passos que descreve uma interação entre um usuário e um sistema” (FOWLER, 2005, p. 104).

Na descrição de um Caso de Uso, os usuários são chamados de atores. Os atores realizam os casos de uso e podem assumir diversos papéis. Ainda segundo FOWLER, “O ator é um papel que o usuário desempenha com relação ao sistema”. Existem várias formas de se descrever um caso de uso. Uma delas é descrever o cenário principal de sucesso através de uma sequência de passos numerados e em seguida descrever suas extensões ou falhas: “Um cenário é uma sequência específica de ações e interações entre atores e o sistema” (LARMAN, 2007, p.89).

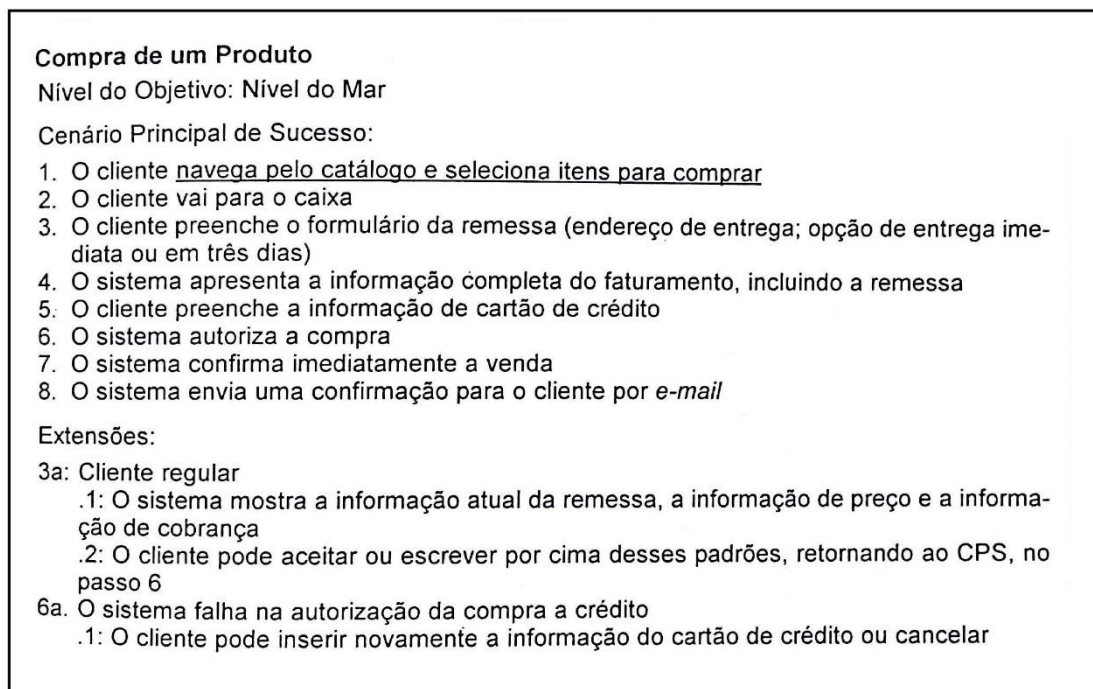


Figura 11 - Exemplo de texto de caso de uso. Fonte: FOWLER (2005, p. 105)

Além disso é possível representar os casos de uso na UML através de um diagrama. O diagrama pode ser muito útil para mostrar os atores, os casos de uso e os relacionamentos entre eles, explicitando O papel de cada ator em cada caso de uso e interações entre casos de uso, ou qual caso de uso inclui outro.

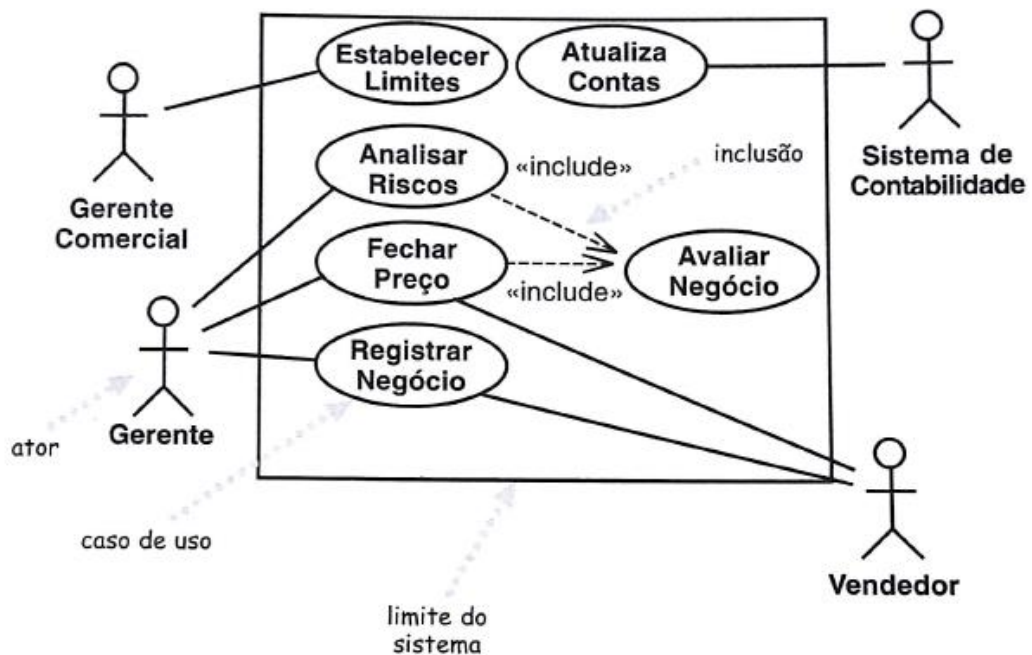


Figura 12 - Diagrama de casos de uso. Fonte: FOWLER (2005, p.107)

Diagrama de Máquinas de Estados

O diagrama de máquina de estados descreve o comportamento de um sistema. Através dele, são apresentados os estados possíveis em um sistema e suas transições, ou seja, quais são as regras a serem aplicadas para que cada estado se altere para outro:

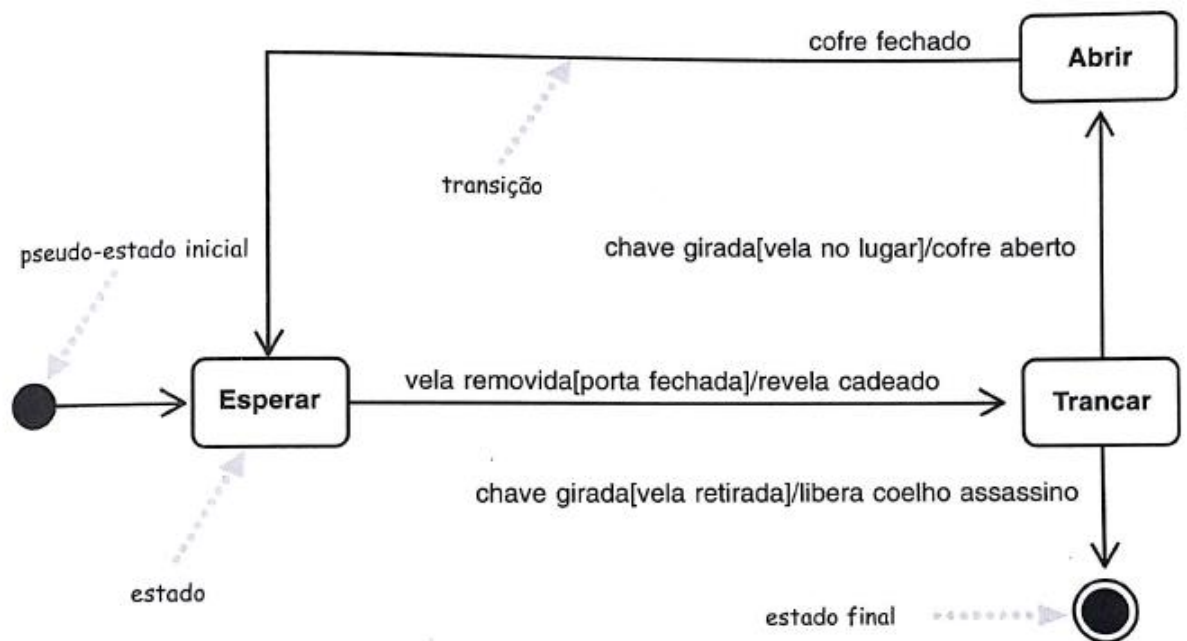


Figura 13 - Um diagrama simples de máquina de estados. Fonte: FOWLER (2005, p. 111)

Diagramas de Interação

Os diagramas de interação são utilizados para descrever as colaborações existentes entre os objetos: “A Interação é um comportamento que compreende um conjunto de mensagens trocadas entre um conjunto de objetos dentro de um contexto para a execução de determinado propósito” (BOOCH, 2012, p.229).

Para melhor desenhar esses comportamentos são criados fluxos de controle que documentam esses aspectos dinâmicos mostrando a sequência a ser seguida além de outros aspectos como ramificações, loops, recursão e concorrência.

Os diagramas de interação são:

- Diagramas de Sequência;
- Diagramas de Comunicação;
- Diagramas de Visão Geral da Interação;
- Diagramas de Sincronização.

Diagramas de Sequência

O diagrama de sequência mostra a troca de mensagens entre objetos dentro de um caso de uso. “Normalmente, um diagrama de sequência captura o comportamento de um único cenário” FOWLER (2005, p. 68). Dessa forma se explicita a interação entre cada objeto do diagrama.

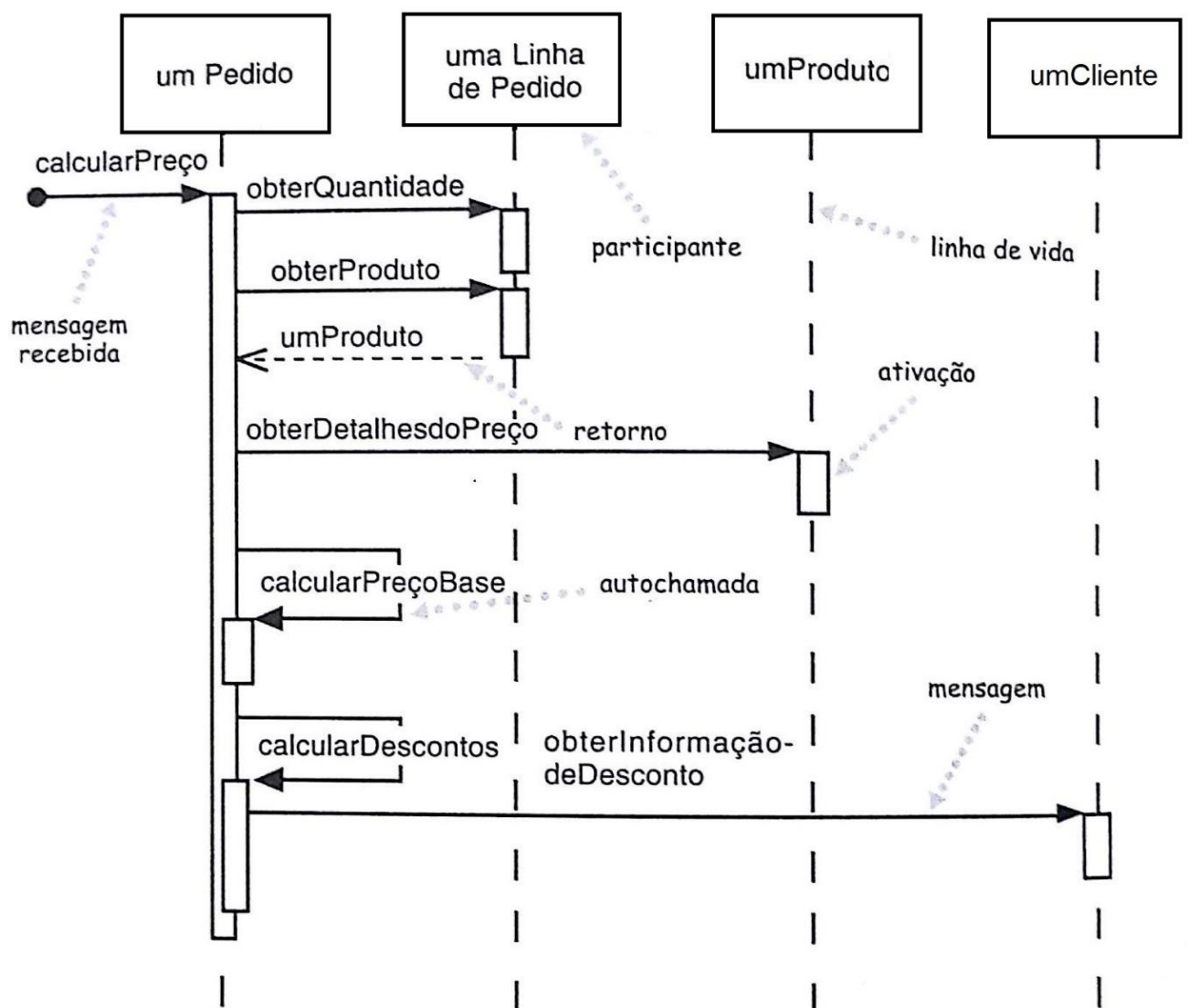


Figura 14 – Um diagrama de sequência para controle centralizado. Fonte: FOWLER (2005, p. 68)

Diagramas de Comunicação

O diagrama de comunicação nomeia um grupo conceitual da arquitetura do sistema. O objetivo é explicitar os vínculos de dados entre os vários atores da interação e pode mostrar os mesmos itens do diagrama de sequência de forma diferente, sendo bem interessante quando o objetivo é explorar alternativas: “os diagramas de sequência são melhores quando você quer salientar a sequência de chamadas e que os diagramas de comunicação são melhores quando quer salientar os vínculos” (FOWLER, 2005, p.131).

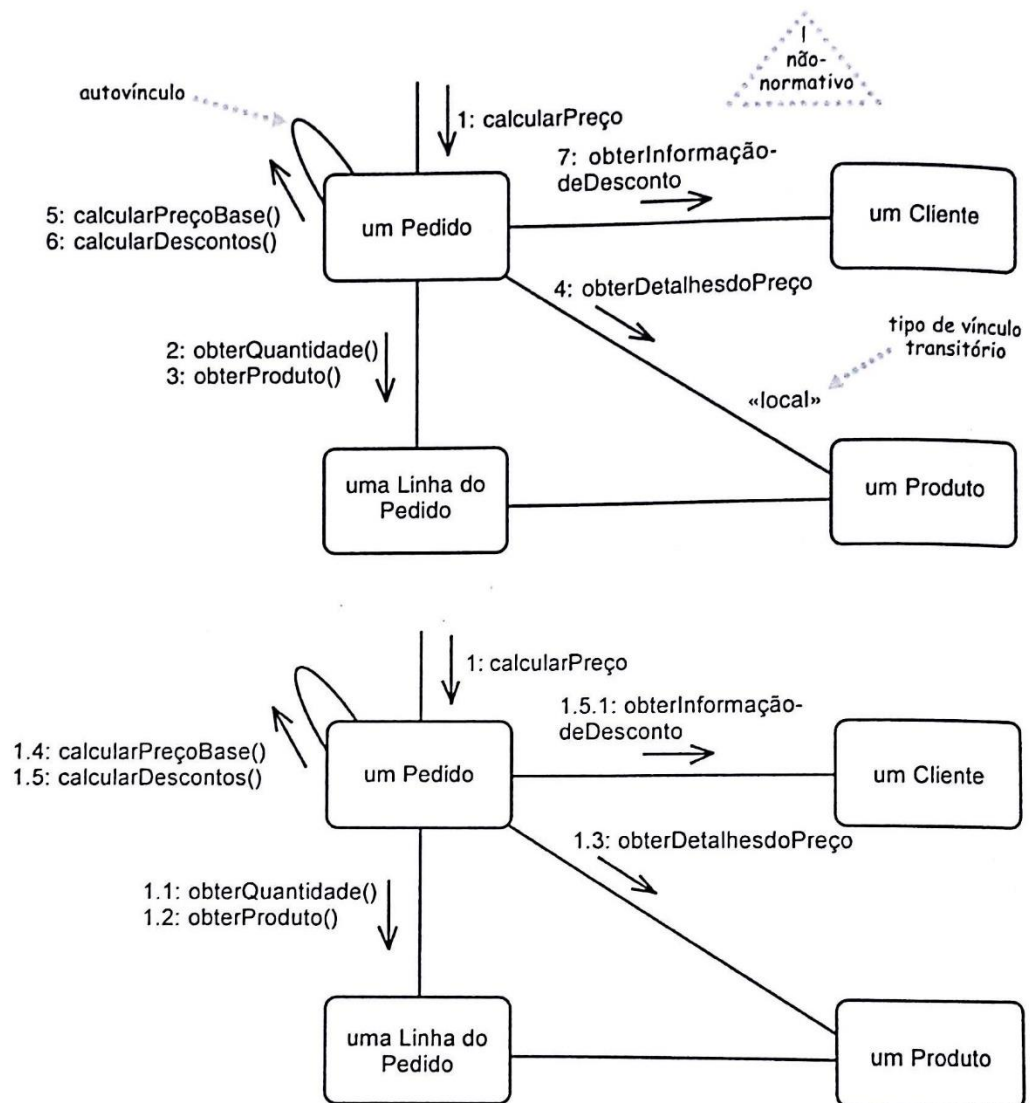


Figura 15 – Um diagrama de comunicação com numeração decimal aninhada. Fonte: FOWLER (2005, p. 130)

Diagramas de Visão Geral da Interação

Os diagramas de visão geral da interação são uma composição de diagramas de atividade e diagramas de sequência. Criados na UML 2, são diagramas de atividade onde cada atividade é substituída por diagramas de sequência aumentando a abrangência do diagrama, mas também podendo se tornar um pouco confuso.

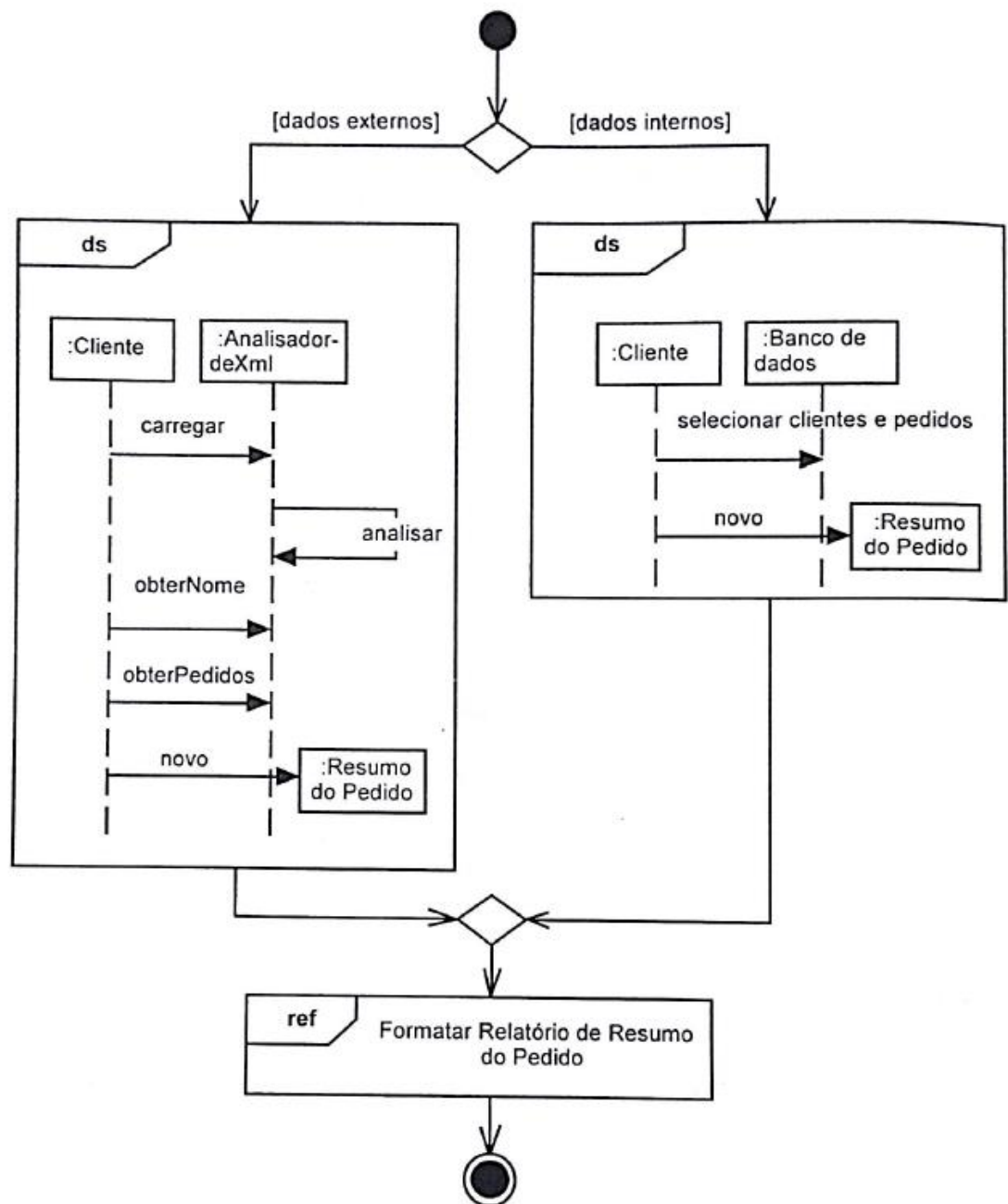


Figura 16 – Diagrama de resumo de interação. Fonte: FOWLER (2005, p. 140)

Diagramas de Sincronização

A sincronização é muito importante para organizar os fluxos de controle de um sistema, a fim de evitar que mais de um fluxo de controle esteja em um objeto simultaneamente, ocasionando erros imprevistos e difíceis de serem localizados e corrigidos. Segundo (BOOCH, 2012, p.360) a UML propõe a modelagem de três soluções distintas:

- **Sequencial** – Quem faz a chamada precisa coordenar externamente o objeto, de maneira que somente um fluxo exista no objeto por vez. Na presença de vários fluxos de controle, a semântica e a integridade do objeto não podem ser garantidas;
- **Protegida** – A semântica e a integridade do objeto são garantidas na presença de vários fluxos de controle pela sequencialização de todas as chamadas para todas as operações protegidas do objeto. Como efeito, exatamente uma operação por vez pode ser invocada ao objeto, reduzindo isso a uma semântica sequencial. Há um perigo de impasse caso não se tome cuidado;
- **Concorrente** – A semântica e a integridade do objeto são garantidas na presença de vários fluxos de controle, porque vários fluxos de controle acessam conjuntos de dados disjuntos ou somente dados de leitura. Essa situação pode ser organizada por regras de projeto cuidadosas.

Através desses diagramas, é possível detalhar todas as interações possíveis dentro de um sistema para validação e comunicação a todas as pessoas da equipe. Segundo (LARMAN, 2007, p.155), os requisitos e interações devem ser organizados por risco, cobertura e criticalidade:

- **Risco** – Inclui tanto a complexidade técnica quanto outros fatores, como incerteza quanto ao esforço necessário ou à usabilidade;

- **Cobertura** – Implica que todas as partes principais do sistema são pelo menos tocadas nas interações iniciais – talvez uma implementação “ampla e superficial” que abranja muitos componentes;
- **Criticalidade** – se refere a funções que o cliente considera de alto valor de negócio.

Segundo Larman, “Esses critérios são usados para priorizar o trabalho ao longo das interações” (LARMAN, 2007, p.156).

Conclusão sobre UML

A UML permite que se faça a modelagem tanto de sistemas simples como de sistemas complexos. Seja programação orientada a objetos ou não, será possível identificar, detalhar e esclarecer o comportamento que o sistema deve apresentar ao final do desenvolvimento e pode ser uma ferramenta de grande ajuda na comunicação dos envolvidos no desenvolvimento de um sistema, na medida em que documenta o entendimento do sujeito que fez a primeira modelagem, e também ao ser compartilhada com as outras pessoas da equipe, que podem contribuir para que o modelo seja cada vez mais preciso e próximo da expectativa do cliente.

BPMN

Outra ferramenta criada para auxiliar a busca por facilitar a comunicação e o entendimento dos processos de uma empresa é a BPMN ou *Business Process Modeling Notation*, uma notação desenvolvida pela Business Process Management Initiative (BPMI), com o objetivo de padronizar os processos de negócios. Através de diagramas, a BPMN descreve todas as etapas de um processo de negócios, coordenando sequências e mensagens que fluem entre os participantes do processo, seja no desenvolvimento de um software ou qualquer outro processo.

No caso do desenvolvimento de software, é uma ferramenta importante para o entendimento das regras de negócios que serão implantadas e também no detalhamento do entendimento do problema a ser resolvido. Por ser uma representação gráfica, permite que todos os participantes falem o mesmo “idioma”, facilitando a comunicação entre os diferentes profissionais envolvidos no processo de desenvolvimento. Além disso, facilita o entendimento com o cliente ou especialista que pode detalhar passo a passo todas as particularidades do processo.

“Um padrão BPMN irá munir os empresários com a capacidade de entender os procedimentos internos de seus negócios em uma notação gráfica e dará às organizações a habilidade de comunicar esses procedimentos de forma padrão. Além disso, a notação gráfica facilitará o entendimento de desempenhos conjuntos e transações de negócios entre organizações promovendo um fácil entendimento entre todos os participantes e possibilitando que os ajustes necessários aos processos sejam feitos rapidamente” (BPMN).

Os diagramas da BPMN são bastante simples para que possam melhorar a facilidade de compreensão dos processos. Cada elemento possui uma representatividade específica para indicar o evento, a atividade, o fluxo ou a colaboração. Vejamos as principais notações abaixo:

Elementos Básicos

A notação BPMN foi criada com o objetivo de ajudar a documentar os processos de negócios de forma que fique claro para todos os envolvidos. Para isso era necessário que a BPMN fosse bastante simples e de fácil entendimento. Assim, foram criados alguns elementos básicos para a composição de diagramas e, a partir desses, algumas variações mais complexas que possibilitam o detalhamento de processos também mais complexos, sem perder a clareza. Ao visualizar um dos elementos de um diagrama, podemos facilmente identificar qual é o elemento básico que o originou. “Dentro das categorias básicas de elementos, podem ser adicionadas variações e informações adicionais para dar suporte aos requisitos de complexidade sem alterar radicalmente a aparência básica do diagrama” (OMG, 2010, p.27).

Os quatro elementos mais básicos são Evento, Atividade, Gateway e Fluxo e, são representados através das formas da figura 17.

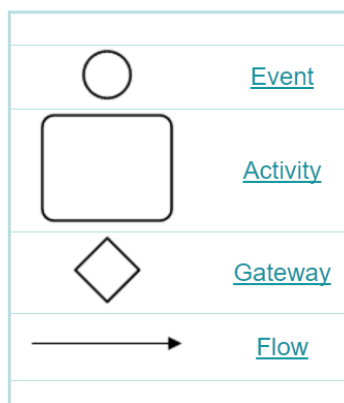


Figura 17 - BPMN Basics. Fonte: Adaptado de OMG(2011)

Eventos

Um evento é algo que "acontece" durante um processo ou uma coreografia. Esses eventos afetam o fluxo do modelo e geralmente têm uma causa (gatilho) ou um impacto (resultado). Os eventos são círculos com centros abertos para permitir que

os marcadores internos diferenciem diferentes gatilhos ou resultados (OMG, 2011, p.29).

Existem três tipos de eventos, conforme figura 18: *Início* (Start), *Intermediário* (Intermediate Event) e *Fim* (End Event). Cada um deles afeta o fluxo de forma diferente.

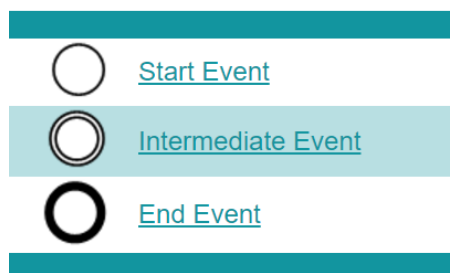


Figura 18 – Elementos representativos de Eventos. Fonte: Adaptado de OMG(2011)

Atividades

Uma atividade é um termo genérico para o trabalho que a empresa executa em um processo. Uma atividade pode ser atômica ou não atômica (composto). Os tipos de Atividades que fazem parte de um Modelo de Processo são: *Subprocesso* (Sub Process) e *Tarefa* (Task), que são retângulos arredondados. As atividades são usadas em processos padrão e em coreografias (OMG, 2011, p.29).

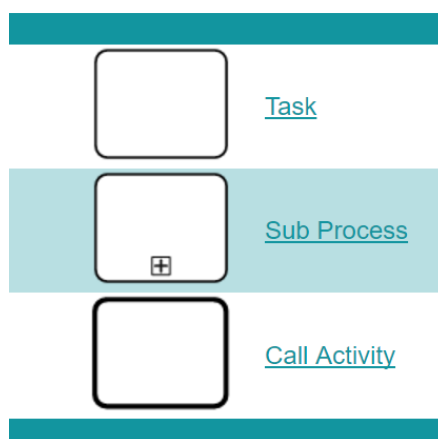


Figura 19 – Elementos representativos de Atividades. Fonte: Adaptado de OMG(2011)

Gateway

Um Gateway é usado para controlar a divergência e a convergência de Fluxos de Sequência em um Processo. Assim, determinará a ramificação, a bifurcação, a fusão e a união de caminhos. Os marcadores internos indicam o tipo de controle do comportamento (OMG, 2011, p.29).



Figura 20 – Elementos representativos de Gateway. Fonte: Adaptado de OMG(2011)

Fluxos

Os fluxos são utilizados para demonstrar a ordem em que as atividades devem ser executadas no processo, o fluxo que as mensagens entre dois participantes que estão preparados para enviá-los e recebê-los deve seguir e ainda pode ser utilizado através de Associações para vincular informações e Artefatos com elementos gráficos BPMN. Anotações de texto, por exemplo, e outros artefatos podem ser associados aos elementos gráficos. Uma ponta de flecha na Associação indica uma direção de fluxo (por exemplo, dados), quando apropriado (OMG, 2011, p.29).

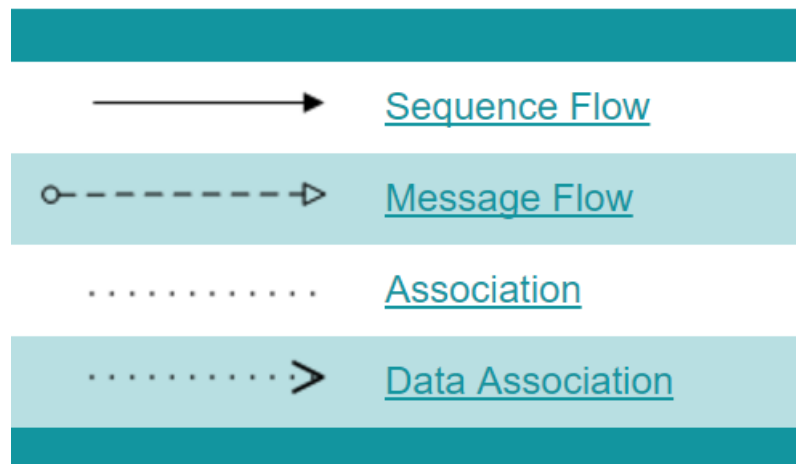


Figura 21 – Elementos representativos de Fluxos. Fonte: Adaptado de OMG(2011)

Dados

Os objetos de dados fornecem informações sobre o que as atividades exigem para serem executadas e/ou o que elas produzem. Objetos de dados podem representar um objeto singular ou uma coleção de objetos (OMG, 2011, p.30).

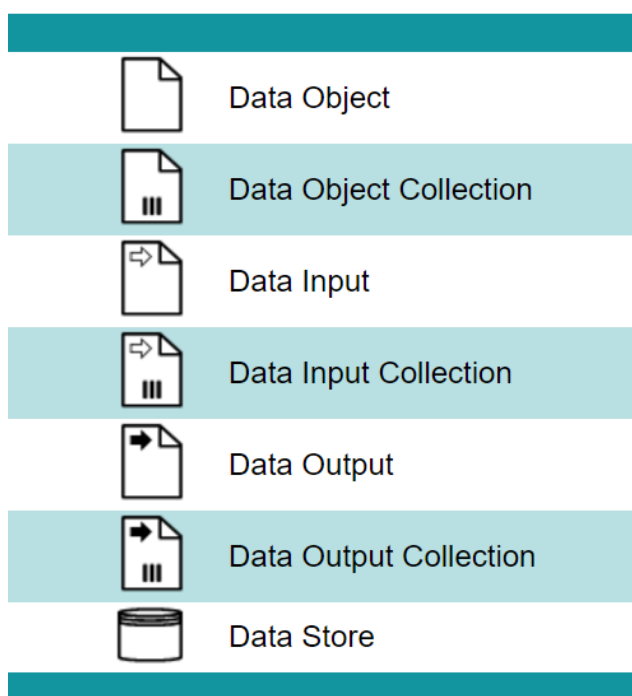


Figura 22 – Elementos representativos de Dados. Fonte: Adaptado de OMG(2011)

Artefatos

Artefatos são elementos adicionais que podem ser incluídos ao diagrama sem afetar o fluxo ou executar qualquer tarefa. Os artefatos podem ser Grupos e Anotações de Texto. Um grupo é um agrupamento de elementos gráficos que estão dentro da mesma categoria. Este tipo de agrupamento não afeta os Fluxos de Sequência dentro do Grupo. O nome da categoria aparece no diagrama como o rótulo do grupo. As categorias podem ser usadas para fins de documentação ou análise. Os grupos são uma maneira pela qual categorias de objetos podem ser exibidas visualmente no

diagrama. Já as anotações de texto são um mecanismo para um modelador fornecer informações de texto adicionais para o leitor de um Diagrama BPMN (OMG, 2011, p.30).

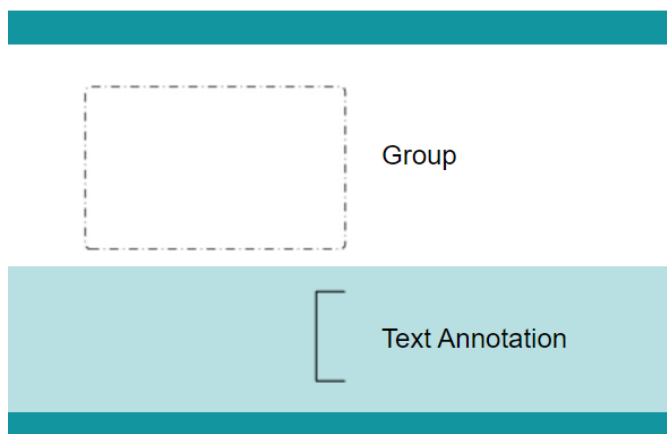


Figura 23 – Elementos representativos de Artefatos. Fonte: Adaptado de OMG(2011)

Colaborações

Colaborações são elementos de organização. Basicamente eles descrevem quem faz o quê.

Um *Pool* é a representação gráfica de um participante em uma colaboração. Ele também atua como um "swimlane" ou raia, e é um recipiente gráfico para particionar um conjunto de Atividades de outros Pools, geralmente no contexto de situações B2B. Uma associação pode ter detalhes internos, na forma do processo que será executado. Ou uma associação pode não ter detalhes internos, ou seja, pode ser uma "caixa preta".

A *Lane* é uma sub-partição dentro de um Processo, ou dentro de um Pool. Quando presente no diagrama estenderá todo o comprimento do Processo, verticalmente ou horizontalmente. Lanes são usados para organizar e categorizar Atividades (OMG, 2011, p.30).

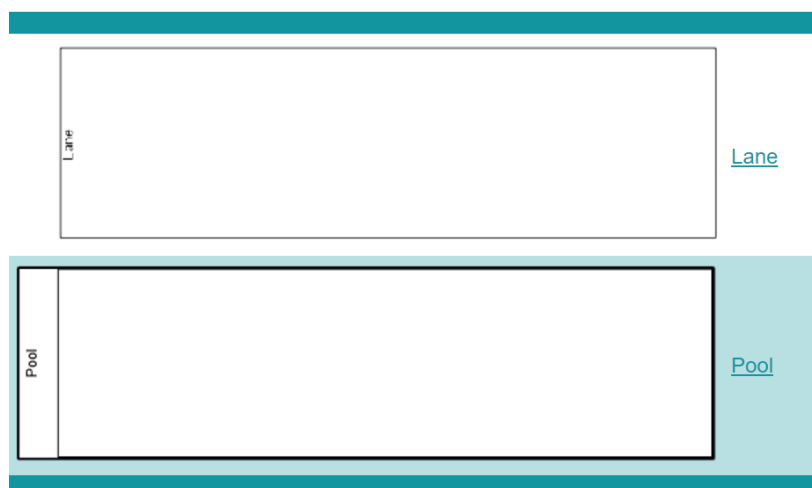


Figura 24 – Elementos representativos de Swimlane basics (Colaboração). Fonte: Adaptado de OMG(2011)

Existem diversos softwares desenvolvidos para facilitar a diagramação dos processos através do padrão BPMN. Alguns exemplos são Bizagi, Heflo, Sydle e Aris.

É importante ressaltar que BPMN e UML não competem entre si. Enquanto a UML é voltada à modelagem para o desenvolvimento de software, BPMN tem seu foco em modelar qualquer processo. O objetivo de ambas é facilitar o entendimento entre pessoas. Explicitar o diálogo em relação ao objetivo a ser conquistado, facilitando a comunicação entre todas as pessoas da equipe e o cliente, que pode perceber processos que não foram comunicados e, muitas vezes, não foram mapeados e fluxos que não foram detalhados da forma como deveriam ter sido feitos.

CONCLUSÕES

Como dito anteriormente, criar software, no sentido do planejamento, não é diferente de criar qualquer outra coisa. Demanda não só um claro entendimento do que se deseja criar, como um bom planejamento e uma metodologia adequada, pois situações semelhantes à da analogia arroz-risoto apresentada acontecem no desenvolvimento de sistemas com uma frequência muito maior do que todos gostariam e existe um forte hábito de tentar transformar arroz em risoto, por custo, tempo, falta de organização ou de conhecimento - o que faz com que seja comprometida a qualidade do sistema entregue.

Não há dúvidas de que a área de Tecnologia da Informação conta com uma grande quantidade de profissionais altamente qualificados. Profissionais dispostos a aprender novas linguagens e técnicas de programação.

Alinhado a isso, temos diversos cursos de graduação e cursos complementares de lógica de programação e linguagens diversas. Além dos cursos presenciais, existe uma tendência cada vez maior de se fazer cursos online, orientados por um professor ou não. Tudo isso contribui para a formação dos profissionais que percebem a necessidade de atualização constante e de se destacarem dos demais profissionais. Em um mercado cada vez mais competitivo, o profissional da área de TI que não acompanha as rápidas novidades que surgem a cada momento facilmente perde espaço e vê sua carreira entrar em declínio.

Vimos aqui que o segmento de desenvolvimento de software por encomenda representa 10% de todo o mercado de TI no Brasil e, entre as pequenas e médias empresas, o segmento de TI é o que mais cresce. Entretanto, mesmo com este cenário promissor, o índice de entrega de projetos no prazo e custo é em geral abaixo de 25%, exceto para pequenos projetos, que atingem índices de 62% de sucesso, segundo o relatório Chaos Report. É fato que projetos menores possuem menos profissionais envolvidos, o que nos leva a crer que as interações entre duas ou mais pessoas interferem negativamente na evolução dos projetos.

Como tentativa de entender os problemas de interação e entendimento entre as pessoas da equipe e também entre a equipe e o seu cliente, verificamos o perfil de cada pessoa através das Inteligências Múltiplas de Howard Gardner, onde pudemos

perceber que os profissionais da área de TI possuem suas inteligências lógico-matemáticas mais desenvolvidas que suas inteligências linguísticas, o que não favorece a comunicação. Verificamos também que, à medida que aumente a sua percepção da importância por desenvolver a inteligência linguística, aliada a inteligência lógico-matemática, o profissional poderá buscar incluir em sua formação habilidades de comunicação. Gardner nos mostra também que todos os indivíduos deveriam desenvolver também suas outras inteligências para que isso favoreça a resolução de problemas diversos. Uma opção, para ajudar a minimizar este problema, seria as universidades incluírem em seus programas mais disciplinas que estimulem a solução de problemas no âmbito teórico e não apenas computacional, além de estimular debates e questionamentos que melhorem as habilidades de comunicação dos futuros profissionais de TI.

Também observamos que através da Gestão de Conhecimento, é possível que os profissionais mais experientes transmitam a informação para os profissionais menos experientes, contribuindo para que estes possam assumir novas tarefas do projeto.

Verificamos que UML e BPMN são ferramentas importantes para o detalhamento do entendimento de um ou mais indivíduos, assim como permitem uma padronização e a documentação de regras de negócios, da sequência em que cada tarefa deve ser executada, exceções que devem ser tratadas e papéis que cada um deve assumir na execução de cada tarefa. Mais do que isso, percebemos que a modelagem de processos nos faz organizar nossos pensamentos e nossas ações e, independentemente da ferramenta utilizada para documentar o processo, este passo é fundamental na estruturação de um projeto, não apenas para facilitar o entendimento entre as pessoas de uma equipe, mas também para que seja armazenado o histórico da evolução do projeto e para documentar quais foram os cenários que levaram à criação de cada etapa, facilitando bastante o trabalho dos profissionais que posteriormente necessitem fazer qualquer manutenção ou reprodução do sistema.

A melhor solução de um problema, formalizada em código fonte, só acontecerá se todos da equipe se percebem, se compreendem e estabelecem o diálogo e a comunicação.

A comunicação é o intercâmbio de percepções entre dois indivíduos e só é efetiva quando essas percepções se fundem, gerando um novo lugar de conhecimento. Ela é fundamental para o ser humano em todas suas áreas e está ligada à percepção do mundo e do outro. Mesmo todas as técnicas que certamente facilitam e organizam a comunicação entre os profissionais de TI e também entre a empresa e o cliente, não poderão resolver o problema da comunicação falha entre as pessoas, pois a verdadeira comunicação se dá ao olhar o mundo com o olhar do outro. Entendemos a nós mesmos quando nos enxergamos através do outro.

Assim, o primeiro passo para melhorar a nossa comunicação talvez seja trabalhar o nosso olhar sobre o outro. Seja um colega de trabalho ou cliente, se conseguirmos perceber o outro, dentro do seu *campo*, conseguiremos nos aproximar da sua visão do mundo e, dessa forma, entender se estamos percebendo um problema da mesma maneira. Isso pode nos levar a entender que a nossa percepção de uma necessidade do projeto, por exemplo, não está alinhada com o que o cliente gostaria. Percebemos isso ao perceber a visão do outro que nos mostra um ponto de vista que não nos pertencia até então.

Segundo Merleau-Ponty, a comunicação ocorre não através do lançamento de ideias, mas sim quando conseguimos perceber a forma como nossas ideias impactaram o outro. Buber diz que, para que a comunicação ocorra, é necessário que estejamos presentes, percebendo o outro. Os dois filósofos nos mostram que a comunicação é uma troca de percepções que ocorre não quando tentamos dividir com o mundo aquilo que acreditamos ser verdade, mas quando observamos a nossa verdade aos olhos do outro, quando o vemos, e ouvimos. Talvez a nossa inteligência linguística, como Gardner classificou, comece não pela articulação de frases e palavras, mas pelo simples ato de ouvir, de observar de perceber.

Quando conseguimos estabelecer uma conexão onde percebemos não só o objeto que o outro olha, mas o que aquele objeto representa para ele, considerando todo o seu *campo*, como Merleau-Ponty chama àquilo que nos faz ser quem somos através de tudo o que já fomos e vivemos. Nesse momento conseguiremos estabelecer esta conexão, a comunicação acontecerá e todas as pessoas irão de fato desenvolver o mesmo projeto. Se todas as pessoas de uma equipe abrissem seus campos, entregando suas ideias e escutando de fato as ideias dos outros, teríamos um grande campo do projeto onde cliente e desenvolvedores teriam a mesma

percepção da real demanda, e ao mesmo tempo seriam tocados, cada um em sua individualidade, compreendendo seu papel e contribuindo para a melhor entrega.

REFERÊNCIAS BIBLIOGRÁFICAS

ABES. **Mercado Brasileiro de Software: panorama e tendências, 2016**, 1ª. Edição. São Paulo: ABES - Associação Brasileira das Empresas de Software, 2016.

BOOCH, Grady e RUMBAUGH, James e JACOBSON Ivar. **UML: Guia do Usuário**, 2ª Edição. Rio de Janeiro: Elsevier. 2012.

BOOG, Gustavo G. e BOOG Magdalena T. **Manual de Treinamento e desenvolvimento: processos e operações**. São Paulo: Pearson Prentice Hall, 2006.

BUBER, Martin. **Eu e Tu. Tradução do Alemão, introdução e notas por Newton Aquiles Von Zuben**, 10ª Edição. São Paulo: Centauro, 2006.

BUBER, Martin. **Do diálogo e do dialógico**, 1ª Edição. São Paulo: Perspectiva, 2014.

CAMPOS, André L. N. **Modelagem de Processos com BPMN**, 2ª Edição. Rio de Janeiro: Brasport Livros e Multimidia, 2014.

CHAVES, Lúcio. **Gerenciamento da comunicação em projetos**, 2ª Edição. São Paulo: FGV, 2010.

DELOITTE e PME, **As PMEs que mais crescem no Brasil: As práticas das empresas emergentes em saúde e bem-estar**, disponível em

<https://www2.deloitte.com/content/dam/Deloitte/br/Documents/conteudos/pmes/PME2014.pdf>. Acessado em 08/06/2017.

FREIRE, Paulo. **Extensão ou Comunicação**, 8ª Edição. Rio de Janeiro: Paz e Terra, 1985.

FOWLER, Martin. **UML Essencial: Um breve guia para a linguagem-padrão de modelagem de objetos**, 3ª Edição. Porto Alegre: Bookman, 2005.

GARDNER, Howard. **Inteligências múltiplas, a teoria na prática**. Porto Alegre: Penso, 2000.

GAVOLI, Eliana. **Uma contribuição para o delineamento do Profissional de TI frente aos papéis exercidos durante o processo de desenvolvimento de Software**, 2009.

GUENGERICH, Steven. **Downsizing em sistemas de informação: reengenharia de sistemas de informação**, 1ª Edição. São Paulo: Makron Books, 1993.

HARASIM, Linda e TELES, Lúcio e TUROFF, Murray e HILTZ, Starr Roxanne. **Redes de Aprendizagem, Um guia para ensino e aprendizagem on-line**, 1ª Edição. São Paulo: Editora Senac, 2005.

LARMAN, Craig. **Utilizando UML e padrões: uma introdução à análise e ao projeto orientado a objetos e ao desenvolvimento iterativo**, 3ª Edição. Porto Alegre: Bookman, 2007.

MARCONDES, Francisco Supino, MONTINI, Denis Ávila, VEJA, Ítalo Santiago, DIAS, Luiz Alberto Vieira. Artigo: **Elicitação da dificuldade de entendimento com base na teoria de Begson.**

MERLEAU-PONTY, Maurice. **A prosa do mundo**, Edição CosacNaify portátil. São Paulo: Cosac Naify, 2012.

MERLEAU-PONTY, Maurice. **Fenomenologia da Percepção**, 4ª Edição. São Paulo: WMF Martins Fontes., 2011.

MERLEAU-PONTY, Maurice. **O Olho e o Espírito**, 1ª Edição CosacNaify portátil. São Paulo: Cosac Naify, 2013.

OLIVEIRA, Djalma de Pinho Rebouças de. **Sistemas de Informações Gerenciais: estratégias, táticas operacionais**, 3ª Edição. São Paulo: Editora Atlas, 1996.

OMG. Business Process Model and Notation (BPMN) Version 2.0, OMG, 2011. Disponível em <http://www.omg.org/spec/BPMN/2.0>. Acesso em 25/06/2017.

PALLOFF, Rena M.e PRATT, Keith. **O Aluno Virtual, Um guia para trabalhar com estudantes on-line**, 1ª Edição. Porto Alegre: Artmed, 2004.

PEREIRA, Alice Cybis (organizadora). **Ambientes Virtuais de Aprendizagem**, 1ª Edição. Rio de Janeiro: Editora Ciência Moderna Ltda., 2007.

SANTAELLA, Lucia. **Comunicação e Pesquisa**, São Paulo: Hacker Editores, 2001.

SEBRAE. **Sobrevivência das empresas no Brasil**, Brasília, SEBRAE, 2016. Disponível em <https://m.sebrae.com.br/Sebrae/Portal%20Sebrae/Anexos/sobrevivencia-das-empresas-no-brasil-relatorio-2016.pdf>. Acessado em 08/06/2017.

TAKEUCHI, Hirota e NONAKA, Ikujiro. **Gestão de Conhecimento**, Porto Alegre: Bookman, 2008.

TARAPANOFF, Kira. **Inteligência, informação e conhecimento**. Brasília: ibict, 2006.

TEIXEIRA, João de Fernandes. **Mentes e máquinas, uma introdução a ciência cognitiva**, Porto Alegre: Artes Médicas, 1998.

YIN, Robert. **Estudo de caso, Planejamento e métodos**, 5ª Edição. Porto Alegre: Bookman, 2015.

Sites

- BMPL Definição – disponível em: <http://xml.coverpages.org/bpml.html>. Acesso em 25 out. 2015.
- <http://www.bpmn.org> Acesso em 14 de set. 2016
- <http://www.devmedia.com.br/introducao-ao-business-process-modeling-notation-bpmn/29892> Acesso em 4 de mar. 2017

Referências

- 1- Retirado do site <http://www.biography.com/people/eli-whitney-9530201>. Acesso em 24 de jun. 2017

- 2- Retirado do site <http://www.biography.com/people/eli-whitney-9530201> Acesso em 24 de jun. 2017

- 3- Retirado do site <http://www.biography.com/people/eli-whitney-9530201> Acesso em 24 de jun. 2017

- 4 - “An analysis of the first federal census demonstrates that from 1790 to the end of the nineteenth century, the percentage of employment dependent on agriculture consistently decreased. For example, in 1790, 75 percent of the workforce consisted of farm laborers, and this rate dropped to about 40 percent near the turn of the twentieth century” Retirado do site <http://career.iresearchnet.com/career-development/industrial-revolution/>. Acesso em 12 de set. 2015

- 5 - Uma boa abordagem da questão das sociedades disciplinares pode ser encontrado em DELEUZE, Gilles: Post-Scriptum sobre as Sociedades de Controle. In Conversações. São Paulo: Editora 34, pp 219 - 26, 1992

- 6 – Retirado do site <https://www.infoq.com/articles/standish-chaos-2015> Acesso em 10 de jul. 2016

- 7 – Tradução livre do original em inglês: A Phenomenology of Skill Acquisition as the basis for a Merleau-Pontian Nonrepresentationalist

- 8- Artigo **A Phenomenology of Skill Acquisition as the basis for a Merleau-Pontian Nonrepresentationalist**
Cognitive Science, Hubert L. Dreyfus, Department of Philosophy, University of California, Berkeley. P.4

- 9- Artigo **A Phenomenology of Skill Acquisition as the basis for a Merleau-Pontian Nonrepresentationalist**
Cognitive Science, Hubert L. Dreyfus, Department of Philosophy, University of California, Berkeley. P.5