

4 Análise dos Resultados

4.1. Introdução

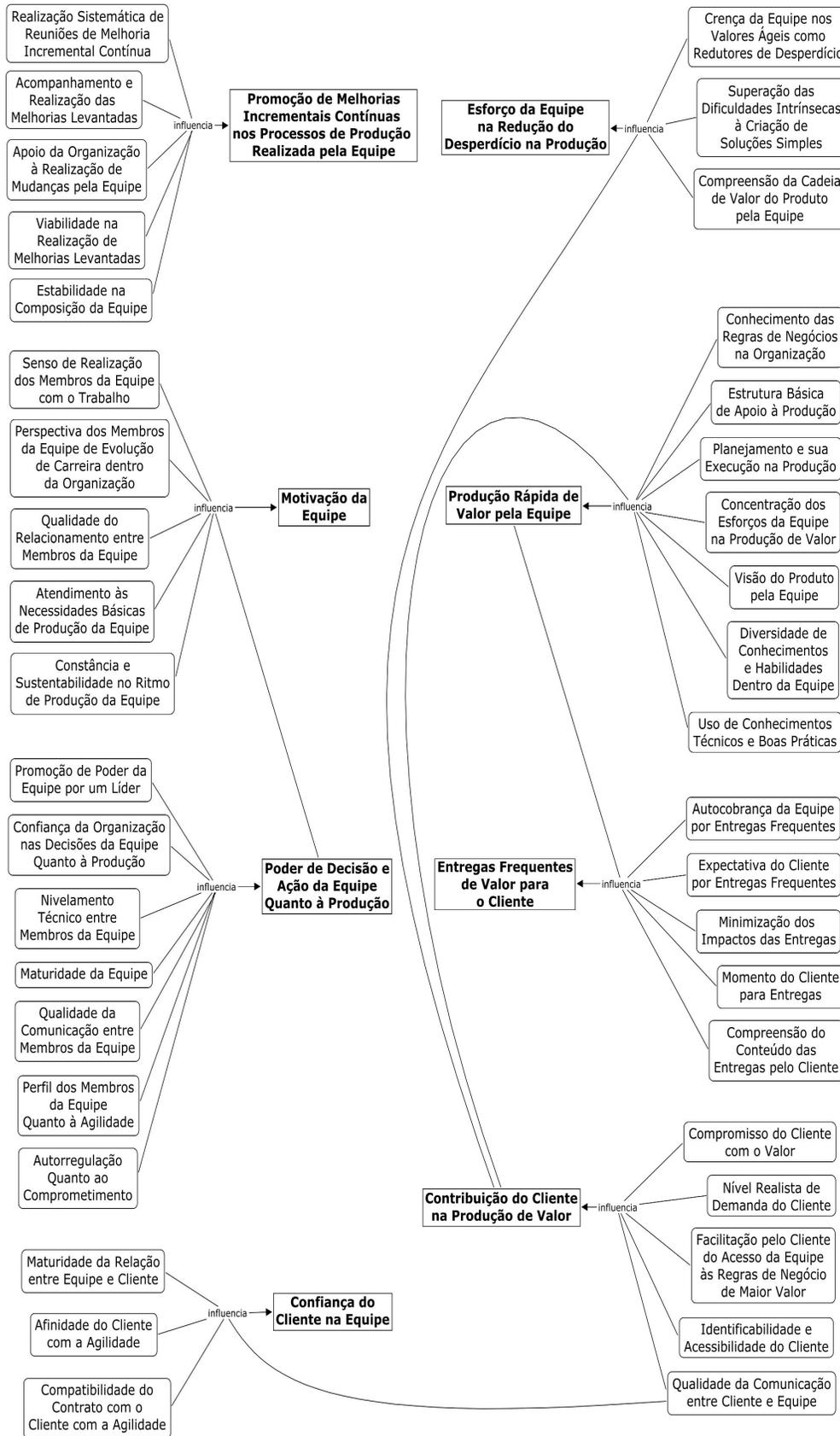
Neste capítulo, procuraram-se identificar, a partir das entrevistas realizadas com *ScrumMasters* e membros de equipes de desenvolvimento de sistemas de tecnologia da informação (TI), os fatores críticos para a prática dos valores Ágeis em equipes que utilizam o *framework* Scrum. Além disso, para cada fator crítico identificado, buscaram-se relacionar as condições que influenciam a manifestação do fator. Cada um desses fatores crítico e condições influenciadoras é explicado e analisado, utilizando-se como base as teorias e os conceitos abordados no capítulo de referencial teórico, com o intuito de fundamentar conceitualmente a análise. Para ilustrar os fatores e condições identificados na análise, citações provenientes das entrevistas são relacionadas aos mesmos.

Na denominação e análise dos fatores críticos e de suas condições influenciadoras correspondentes, uma equipe de desenvolvimento de sistemas de TI que utiliza o *framework* Scrum é chamada neste trabalho de “equipe” (e é também chamada de “time” por alguns entrevistados), um membro de uma equipe é chamado de “membro”, o líder de uma equipe é chamado de “*ScrumMaster*”, a empresa ou organização fornecedora em que está inserida uma equipe é chamada de “organização” e a empresa ou organização contratante de projetos de desenvolvimento de *software* de uma organização em que está inserida uma equipe é chamada de “cliente”. Estão também incluídos na denominação de “cliente” quaisquer representantes dessa empresa ou organização contratante, que geralmente atuam no papel de *Product Owners*, indistintamente quanto a serem contratados ou designados para o papel pelo próprio cliente ou pela organização (fornecedora). Uma observação importante é que, em casos em que não há cliente externo para o projeto (como é, por exemplo, para um projeto interno ou para um produto desenvolvido para ser lançado no mercado), o *Product Owner* também assume o papel de cliente. E, por fim, qualquer usuário final do produto criado

pelo projeto, pertença ele ao cliente ou não, é chamado de “usuário” neste trabalho.

Para possibilitar a distinção entre entrevistados nas citações das entrevistas, serão utilizadas as mesmas definições de “organização”, “equipe”, “membro” e “*ScrumMaster*” descritas acima, com a adição de numeração sequencial quando for necessária a diferenciação entre membros de uma mesma equipe, entre equipes de uma mesma organização e entre organizações diferentes, no seguinte formato: Organização X/Equipe Y/Membro Z ou *ScrumMaster*, onde X, Y e Z representam a identificação unitária.

A análise do material obtido por meio das entrevistas resultou na identificação de oito fatores críticos que ordenam o presente capítulo. Esses fatores e suas respectivas condições influenciadoras são sintetizados no quadro 3. Vale destacar que alguns fatores críticos funcionam como condições que influenciam a manifestação de outros fatores críticos.

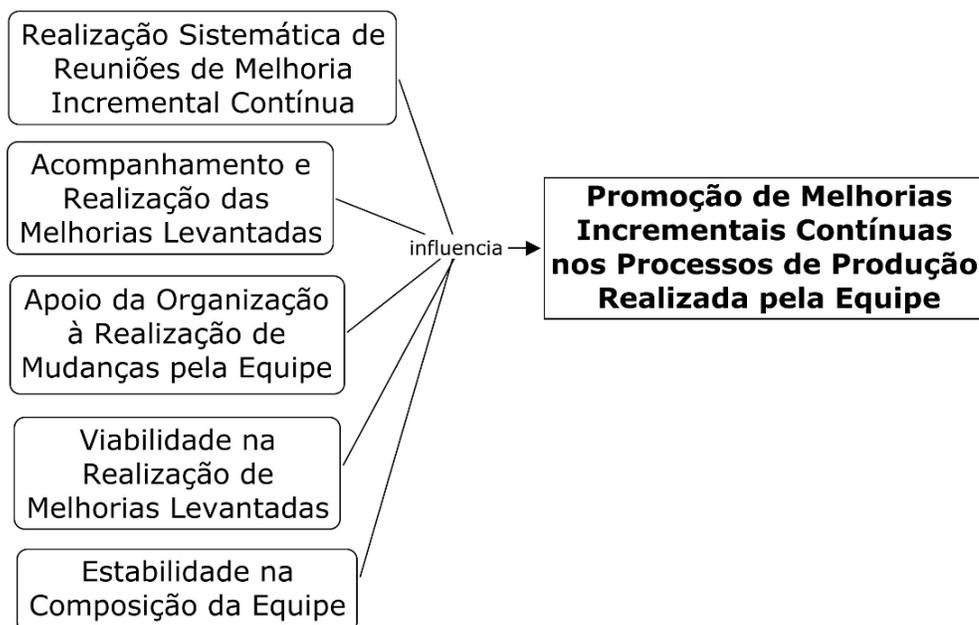


Quadro 3: Fatores críticos e condições que influenciam sua manifestação

4.2. Fatores Críticos para a Prática de Valores Ágeis

Os tópicos seguintes apresentam e analisam cada um dos oito fatores críticos para a prática dos valores Ágeis utilizando-se o *framework* Scrum, identificados a partir das entrevistas realizadas com os membros e os *ScrumMasters* das equipes de desenvolvimento de sistemas de tecnologia da informação (TI). Para cada fator crítico, são apresentadas as condições que influenciam a manifestação do fator.

4.2.1. Promoção de Melhorias Incrementais Contínuas nos Processos de Produção Realizada pela Equipe



Quadro 4: Fator crítico “promoção de melhorias incrementais contínuas nos processos de produção realizada pela equipe” e condições que influenciam sua manifestação

A equipe promover melhorias incrementais contínuas nos processos de produção, chamadas de *kaizen* na Produção Enxuta (WOMACK & JONES, 1998), aparece nas entrevistas como um fator crítico para a prática dos valores Ágeis. Esse resultado já era esperado, visto que a melhoria incremental contínua é explicitada nas perguntas das entrevistas através do princípio Ágil “em intervalos

de tempo regulares, a equipe reflete sobre como se tornar mais eficaz, para então aprimorar e ajustar seu comportamento de acordo” (FOWLER & HIGHSMITH, 2001), onde ficam caracterizadas tanto a inspeção quanto a adaptação, pilares do Scrum (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). Conforme mencionado por Highsmith (2004), a melhoria incremental contínua nos processos de produção é necessária como uma resposta das organizações à mudança para que possam sobreviver e prosperar na economia atual, de forma que é crítica para a prática do valor Ágil “responder a mudanças mais que seguir um plano”.

A análise das entrevistas indica que as condições listadas a seguir influenciam a promoção de melhoria incremental contínua nos processos de produção. O quadro 4 sintetiza essa relação.

- Realização Sistemática de Reuniões de Melhoria Incremental Contínua
- Acompanhamento e Realização das Melhorias Levantadas
- Apoio da Organização à Realização de Mudanças pela Equipe
- Viabilidade na Realização de Melhorias Levantadas
- Estabilidade na Composição da Equipe

4.2.1.1.

Realização Sistemática de Reuniões de Melhorias Incrementais

A realização sistemática de reuniões de melhoria incremental é concretizada no Scrum através das reuniões de retrospectiva, realizadas ao final de cada *sprint*, ou ciclo de desenvolvimento (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). Da mesma forma, na Produção Enxuta, os eventos de *hansei* (que significa autorreflexão implacável e obrigatória) ajudam a viabilizar o *kaizen*. Esses eventos são reuniões da equipe para a promoção de melhorias incrementais e, quando realizados periodicamente (LIKER, 2003), podem ser considerados equivalentes às reuniões de retrospectiva do Scrum. Sobre esse assunto, Melnyk et al. (1998) sugerem que eventos de *kaizen* sejam realizados de forma frequente e regular para promover de fato as melhorias incrementais e contínuas na organização.

Conforme descrito por Schwaber & Beedle (2002) e Schwaber (2004), a reunião de retrospectiva da *sprint* tem o objetivo bem definido de realização de inspeções e planejamento de adaptações necessárias aos processos de desenvolvimento de *software* da equipe. No contexto da Produção Enxuta, Farris et al. (2009) defendem que essa clareza nos objetivos dos eventos de *kaizen* leva, em última instância, a uma maior motivação da equipe em participar de atividades futuras de melhoria. Nessa mesma linha, Farris et al. (2009) afirmam ainda que eventos de *kaizen* com um escopo e objetivos mais limitados têm mais chances de obter os resultados de negócios desejados.

A importância da realização sistemática dessas reuniões como facilitador à prática de valores Ágeis é explicitada em diversos depoimentos:

“O que facilita é o próprio processo, não é? Que prevê isso já no final de cada *sprint* e a gente faz, seguindo o processo. Seguimos à risca, fazemos retrospectiva, e o fato de fazer retrospectiva, a gente consegue discutir sobre o que foi feito, o que foi abordado, o que pode ser feito, a gente consegue deixar mais claro para todo mundo” (Organização 7/Equipe/Membro).

“O que facilita é realmente a gente fazer essa retrospectiva. A gente separa pontos que a gente precisa melhorar e pontos que deram certo.” (Organização 3/Equipe/Membro).

"Boa parte do quebra-pau que a gente teve que provocou algumas mudanças foi nas retrospectivas" (Organização 1/Equipe 1/Membro 1).

No entanto, segundo entrevistados, indivíduos não acostumados com o nível de exposição necessário para uma participação efetiva nas reuniões de retrospectiva tendem a se sentir intimidados e a evitar enfrentamentos, não apontando até mesmo problemas bem visíveis. Esse problema se acentua na presença de pessoas que têm poder sobre o emprego do indivíduo, o que acontece, por exemplo, quando sócios da empresa fazem parte da equipe de desenvolvimento. Liker (2003) afirma, no entanto, que tratar abertamente do que não deu certo (através do *hansei*) e assumir a responsabilidade é um sinal de força do indivíduo, e não de fraqueza. Esses problemas são relatados por alguns entrevistados:

“As pessoas (...) não tinham uma coragem de levantar o dedinho e falar ‘olha, eu não gosto disso’ (...), uma dificuldade das pessoas colocarem sua opinião. É porque as pessoas tem medo de serem negadas (...) e é até uma questão de adaptação; se você chega em uma empresa diferente e essa empresa tem uma outra forma de

trabalhar, você vai tentar ao máximo se adaptar àquela forma de trabalho, não interessa se você concorda com aquilo ou não” (Organização 1/Equipe 1/Membro 1).

“No começo, a nossa equipe tinha um pouco de receio de falar as coisas erradas, até pela intimidação dos sócios estarem participando do projeto. Então isso foi um paradigma um pouco difícil de quebrar na cabeça deles, para eles entenderem que poderiam (...) criticar sem problemas qualquer coisa que foi feita e que isso não faria ele ser mandado embora. Eles estavam acostumados com outro tipo de ambiente nos projetos, então não batiam de frente com o gerente e nem com outras pessoas. (...) Mesmo escrevendo *post-it* do que foi bom e do que foi ruim, (...) não falavam diretamente ou às vezes deixavam de falar uma coisa que todo mundo sabia que estava errado” (Organização 2/Equipe/*ScrumMaster*).

Alguns *ScrumMasters* entrevistados afirmam que realizam mudanças criativas na dinâmica dessas reuniões como forma de estimular ou melhorar a sua participação, o que inclui desde diferentes técnicas até a realização das mesmas em um bar bebendo cerveja ou em um café. Os seguintes relatos ilustram essas práticas:

“Eu gosto de tentar de vez em quando mudar a dinâmica da retrospectiva. Porque se a gente for seguir sempre o mesmo roteiro durante as reuniões, acho que ela deixa de ser efetiva. Então (...) algumas vezes a gente faz uma discussão verbal, ou a gente usa aquele gráfico de espinha de peixe, ou a gente faz alguma coisa diferente de vez em quando para poder dar uma sacudida no pessoal. Às vezes sai da empresa e vai para outro lugar, vai para um café qualquer para discutir os problemas e eu acho que isso ajuda, que isso facilita” (Organização 10/Equipe/*ScrumMaster*).

“A retrospectiva pode ser feita de várias formas. Já fiz retrospectivas de forma canônica e não canônica. Tipo, no Bracarense [bar carioca] já teve uma retrospectiva. Especialmente depois de ter tido uma vitória fantástica num cliente que tinha uma barreira enorme. (...) As formais acontecem também” (Organização 14/Equipe/*ScrumMaster*).

Em outros casos, dinâmicas ou jogos realizados fora das reuniões também buscam obter o mesmo efeito de aproximar as pessoas e melhorar a sua participação nas reuniões de retrospectiva, como se pode ver abaixo:

“Sabe aquele jogo Imagem e Ação? A gente começou a jogar isso na empresa depois do almoço. Então a gente praticava para fazer com que as pessoas, através das mímicas - a gente usava mímicas - pudessem se expressar melhor e enturmar melhor com a equipe. Isso foi legal porque trouxe um momento de descontração (...) e as pessoas começaram a ficar mais soltas nas reuniões. (...) Tudo isso ajudou a gente a melhorar o resultado da reunião de retrospectiva, das pessoas

expressarem melhor os problemas, de ter uma transparência melhor” (Organização 2/Equipe/*ScrumMaster*).

Um dos *ScrumMasters* entrevistados relata, em seguida, que a equipe realiza diversas reuniões menores de retrospectiva ao longo da *sprint*, ao invés de uma grande em seu final, de forma a colocar as melhorias em prática mais rapidamente:

“A gente acaba fazendo retrospectivas bem pequenas, uma no meio da semana e uma no final da semana. Justamente para a gente já saber a dimensão dos itens que estão sendo levantados e como é que a gente pode melhorar isso de forma mais rápida, sem ter que esperar duas semanas [duração da *sprint* da equipe] para a gente começar a tomar atitude relacionada a isso. (...) Facilita porque a gente consegue justamente colocar as melhorias de forma mais rápida, não fica aquele monte de melhoria que foi acumulada ao longo de duas semanas” (Organização 12/Equipe/*ScrumMaster*).

De forma inversa, algumas equipes realizam as reuniões de retrospectiva de forma mais esparsa e muitas vezes deixam de realizá-las. Mas, como se pode ver nos relatos abaixo, esse comportamento é sempre colocado como um problema:

“Já aconteceu de não ter feito retrospectiva por causa da correria. Nem porque a gente falou ‘vamos fazer’ e ‘não vamos fazer’. Foi porque a gente esqueceu, de tanta pressão. Você sair do planejamento e entrar programando. Correria, isso prejudica bastante. E foi nocivo não ter feito, porque as pessoas acumularam coisas” (Organização 14/Equipe/*ScrumMaster*).

“O problema da retrospectiva é no plano de ação. Muitas vezes não deu tempo de uma *sprint* para outra para atuar naquilo. Então a gente tem feito uma regularidade de a cada um mês e meio a retrospectiva. (...) Talvez não tenha a maturidade para atuar com mais urgência nos problemas. Então talvez seja algo que a gente tenha que melhorar mesmo. Aumentar a frequência e dar a atenção devida” (Organização 4/Equipe/*ScrumMaster*).

“Nesse último *sprint*, (...) a gente não conseguiu fazer essa reunião [de retrospectiva], por exemplo. Alguns *sprints*, a gente acaba não fazendo por falta de tempo. Infelizmente, como essas lições aprendidas são feitas com toda a equipe, com o *Product Owner*, com o desenvolvimento, teste, tudo, (...) o grande problema está em conseguir juntar todas essas pessoas” (Organização 4/Equipe/Membro).

Os entrevistados, assim, atribuem a não realização das retrospectivas à pressão de tempo, à falta de disponibilidade de pessoas que julgam essenciais na reunião ou à possível falta de maturidade da equipe para entender a importância dessas reuniões e da própria mudança. Essa falta de crença no valor da mudança e nos resultados das reuniões de melhorias contínuas, de acordo com o estudo de

Farris et al. (2009), afeta negativamente o desenvolvimento da capacidade de resolução de problemas da equipe.

4.2.1.2.

Acompanhamento e Realização das Melhorias Levantadas

As reuniões sistemáticas de melhorias incrementais contínuas, ou reuniões de retrospectiva (também chamadas de “lições aprendidas” por alguns entrevistados), permitem a realização da inspeção, um dos pilares da teoria de controle de processos empíricos e, portanto, do Scrum. Outro pilar importante é a adaptação, ou seja, o acompanhamento e a realização propriamente dita dos pontos a melhorar detectados nas reuniões de retrospectiva (SCHWABER & BEEDLE, 2002; SCHWABER, 2004).

A visibilidade dos pontos detectados a melhorar aparece nas entrevistas como um aspecto importante para facilitar o acompanhamento das melhorias necessárias detectadas e aumentar o compromisso de seus responsáveis com a sua realização. A importância desse aspecto é compatível com a afirmação de Liker (2003), citada anteriormente, em que tratar-se abertamente do que não deu certo é um sinal de força do indivíduo, e leva a que se assumam as responsabilidades e a que se tomem medidas para resolver as questões levantadas.

Essa visibilidade é geralmente promovida nessas equipes através de quadros ou papéis afixados nas paredes do ambiente de trabalho da equipe, indicando os pontos a melhorar, como se pode ver nos depoimentos a seguir:

“O que a gente precisa melhorar fica numa parede, num quadro que a gente olha. Fica no nosso setor, então está sempre todo mundo de cara com aquilo ali. E cada um da equipe fica responsável... as pessoas se disponibilizam e durante o próximo *sprint* essa pessoa vai ficar responsável para que aquilo não aconteça, quer dizer, ficar monitorando aquilo, inspecionando para realmente aquilo não acontecer novamente.” (Organização 3/Equipe/Membro).

“Geralmente tem as coisas que a gente se compromete a fazer na próxima *sprint* e bota lá no quadro, a gente prioriza o que é mais importante, a gente começa a fazer e a gente se compromete. (...) E deixa lá no quadro até que aquilo ali já esteja no processo normal, e parte para a próxima, e parte para a próxima” (Organização 7/Equipe/Membro).

Mesmo quando essa abordagem não é utilizada, ela é geralmente vista de forma positiva pelos entrevistados:

“[O conjunto de problemas detectados na reunião de retrospectiva] não é algo que fica exposto, por exemplo, no *taskboard* [quadro físico de tarefas]. Talvez, se ficassem, os problemas seriam resolvidos mais rapidamente” (Organização 15/Equipe/Membro).

No entanto, quando não se realiza o registro dos pontos a melhorar no momento em que são detectados (mesmo que durante a *sprint*), eles muitas vezes acabam esquecidos no momento da reunião de retrospectiva:

“Não ter anotado algumas coisas que eu queria ter levado para a reunião da *sprint* e no dia que chegou na reunião da *sprint*, eu não lembrei. Foi junto com as *sprints* mais longas, *sprints* de quatro semanas assim. Quatro semanas é uma vida, não é? O problema acontece, soluciona, você esquece e chega na retrospectiva e às vezes você não consegue levantar” (Organização 8/Equipe/Membro).

Uma vez detectados os pontos de melhoria e distribuídas as responsabilidades pela execução dos planos de ação respectivos entre os membros da equipe e o *ScrumMaster*, esses planos de ação devem ser colocados em prática por seus responsáveis, como relata um dos membros de equipe entrevistados:

“Após levantarmos os problemas [nas reuniões de retrospectiva], tentamos classificar se é responsabilidade do time – *ScrumMaster*, ou do P. O. - empresa para tentar resolver. Nos *retrospectives* seguintes, voltamos nesta lista e vemos se os problemas continuam ocorrendo. Os problemas são resolvidos. Talvez isto ocorra por parte do *ScrumMaster* e pelo time, mas é algo natural” (Organização 15/Equipe/Membro).

Mas em diversos casos a pressão do tempo funciona como um obstáculo para que a equipe trate das melhorias detectadas, que muitas vezes demandam mais tempo do que ela consegue ou está disposta a despende para realizá-las, como reconhece um *ScrumMaster* entrevistado:

“O que dificulta é exatamente ter um ritmo acelerado, a gente tem que ter uma velocidade alta. Acho que ter uma velocidade muito alta dificulta e ter pouco tempo ocioso para olhar isso [as melhorias necessárias]. (...) Na hora do *sprint*, está todo mundo muito focado na meta. Enquanto as coisas da meta não estão totalmente atendidas, o pessoal fica muito focado nisso” (Organização 3/Equipe/*ScrumMaster*).

Em outras equipes, as melhorias necessárias detectadas pela equipe acabam por não ser realizadas, como ilustra o seguinte relato de um *ScrumMaster*:

“Eu tenho sentido que a equipe não participa com tanto empenho na reunião de retrospectiva quanto eu gostaria. Não sei se é porque algumas reclamações que eles tem tido a gente não tem conseguido responder a contento e vê, ‘essa reunião está sendo inútil, eu estou querendo isso aqui, isso não está acontecendo, então para que eu estou fazendo essa reunião?’ (...) Se a reunião não dá resultado, se tem o *inspect* [inspeção] mas não tem o *adapt* [adaptação], então não adianta muita coisa” (Organização 10/Equipe/*ScrumMaster*).

Nesses casos, as reuniões de retrospectiva tendem a perder o sentido para seus membros, e eles se sentem desestimulados a realizá-las.

4.2.1.3.

Apoio da Organização à Realização de Mudanças pela Equipe

O apoio da alta gerência da organização à equipe na realização de mudanças a partir dos pontos a melhorar levantados emerge das entrevistas como uma condição que influencia a promoção de melhorias incrementais contínuas pela equipe. A influência dessa condição está de acordo com as afirmações de Doolen et al. (2008), para quem o apoio da alta gerência a eventos de *kaizen* é importante para que se obtenham resultados iniciais positivos (embora seja necessário acompanhamento posterior), e com as afirmações de Farris et al. (2009), para quem o apoio da alta gerência ao trabalho da equipe afeta a atitude da equipe em direção à promoção de melhorias. Esse apoio da alta gerência na realização de mudanças pela equipe geralmente reflete uma disposição da própria organização quanto às mudanças, como se pode ver no depoimento a seguir:

“Como a cultura da empresa é muito aberta à mudança, uma vez que a gente identificou os itens, é muito rápido ter o diálogo para ver como a gente vai resolver. Não tem um processo burocrático de, por exemplo, ‘a máquina do desenvolvedor X está com problema, vamos pedir para o setor de compras comprar e tal’, esperar um tempão. Às vezes, se você está numa estrutura maior e mais burocrática, as coisas demoram muito para serem resolvidas. (...) A cultura da empresa e o pessoal se sentir com poder de fazer as coisas é muito importante para conseguir adaptar bem às necessidades do projeto e do cliente específico. (...) Tem uma abertura muito grande para a inovação e para trazer ideias aqui. Tem alguns ambientes que são muito fechados para a mudança. Para você conseguir implantar alguma ideia nova, você tem que pedir a aprovação de ene pessoas e você tem que falar com seu gerente, com o diretor. Aqui, se você for no café conversar dez minutos com o sócio da empresa, ele vai falar: ‘que ideia legal, vamos botar para a frente’. Não tem barreiras corporativas que impeçam a mudança” (Organização 14/Equipe/Membro).

Em alguns casos, o apoio organizacional à tomada de decisões por parte da equipe quanto a seus processos de produção não existe. No exemplo a seguir, há interferências importantes nas decisões da equipe, que acabam por comprometer a realização de melhorias contínuas e a autogerência da equipe:

“[O *Product Owner*] tenta se meter demais nas decisões da equipe (...), qualquer tipo de decisão em relação a como a gente acha melhor fazer alguma coisa no Scrum, às vezes ele se sente mais parte da equipe do que deveria ser, de certa forma. Ele tenta influenciar demais as nossas decisões. (...) Às vezes tem discussões também demasiadamente grandes durante as reuniões que deveriam ser problema da equipe” (Organização 1/Equipe 1/Membro 2).

Em outros casos, como relatado a seguir, a alta gerência não apoia sequer o uso do tempo na realização de reuniões de melhorias contínuas:

“O gerente quer que o pessoal esteja lá na mesa, desenvolvendo. O gerente externo. Então (...) o cara queria que eu pulasse o ‘lições aprendidas’ ou fizesse meia boca nas lições aprendidas para já começar a parte de desenvolvimento do próximo *sprint*. Eu forcei a barra mesmo” (Organização 6/Equipe/*ScrumMaster*).

Nessa situação, só se torna possível que reuniões de melhorias contínuas aconteçam quando há persistência da liderança Ágil e da própria equipe.

4.2.1.4. Viabilidade na Realização de Melhorias Levantadas

Alguns pontos a melhorar detectados estão fora do alcance da organização ou da equipe, ou são de difícil realização. Alguns depoimentos de entrevistados mostram situações em que isso acontece, como se pode ver em seguida:

“Como não é uma empresa tão grande, não tem um faturamento tão grande, às vezes algumas soluções não são tão viáveis. Por exemplo, seria legal se a gente tivesse um *datacenter* com um monte de máquinas parrudas para atender tudo e todo mundo ter uma máquina rápida para trabalhar o tempo todo. Isso não é possível sempre, a gente tem que ir no ritmo que as coisas conseguem andar” (Organização 14/Equipe/Membro).

“Tinha muita coisa que não dava para a gente mudar. A gente tinha muita dependência do cliente. O fato de desenvolver no ambiente do cliente, por exemplo. Muitas melhorias tinham que ser feitas lá” (Organização 13/Equipe/Membro).

Enquadram-se nessa categoria, como se podem identificar nos relatos acima, melhorias que exigem um esforço financeiro maior do que a empresa pode ou está disposta a despende e dependências externas do projeto sobre as quais nem equipe, nem o *ScrumMaster* ou qualquer pessoa da organização têm controle direto.

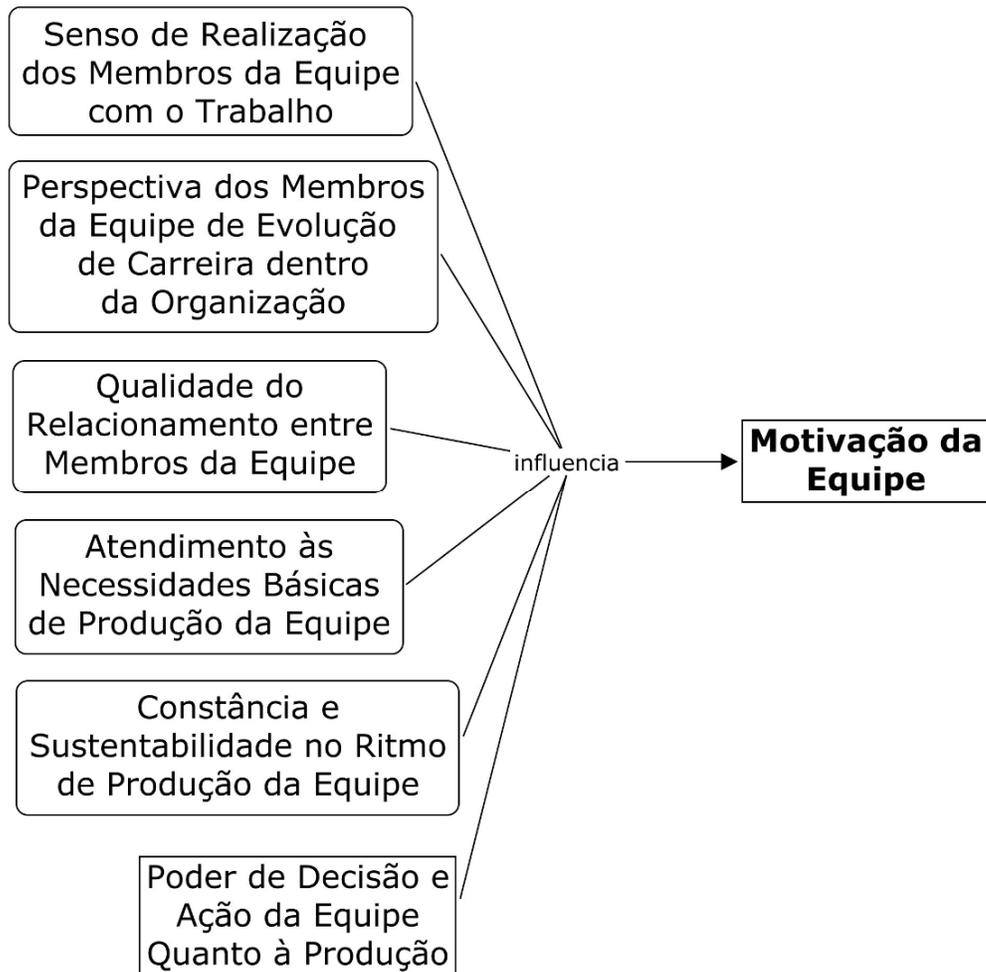
4.2.1.5. Estabilidade na Composição da Equipe

Segundo Cohen (1993), a rotatividade de seus membros é um fator importante para a efetividade de equipes autogerenciadas. Um membro de equipe entrevistado afirma que sua equipe já teve problemas com a alta rotatividade na organização:

“Teve um tempo que a rotatividade na empresa em geral estava grande. Tinha muita gente entrando e saindo, entrando e saindo e isso, no final das contas, prejudicava a passagem do conhecimento. Como a equipe já estava com aquilo dentro, quando entrava uma pessoa nova, essa pessoa nova não tinha aquilo, aquilo já não estava mais no quadro, e tinha que passar de novo por aquele processo” (Organização 7/Equipe/Membro).

Uma baixa rotatividade – e conseqüentemente uma composição da equipe mais estável – permite a retenção do conhecimento adquirido sobre seus processos de produção dentro da equipe, que aprende, como unidade, melhores formas de realizar seu trabalho. Quando há alta rotatividade, essa unidade é quebrada e os conhecimentos e melhorias conquistadas tendem a se dissipar, de forma que a entrada de novos membros cria uma nova unidade que deve passar por seus próprios processos de melhorias contínuas.

4.2.2. Motivação da Equipe



Quadro 5: Fator crítico “motivação da equipe” e condições que influenciam sua manifestação

Os relatos dos entrevistados indicam que a motivação dos indivíduos das equipes de desenvolvimento na realização de seu trabalho é um fator crítico para a prática dos valores Ágeis. O valor Ágil “indivíduos e interações mais que processos e ferramentas”, em particular, reconhece a importância desses indivíduos como quem de fato gera os produtos e serviços (HIGHSMITH, 2004), e assim essas atividades de produção dependem para seu sucesso de questões humanas (LARMAN, 2003) como a motivação. A motivação aparece explicitamente nas perguntas das entrevistas através do princípio Ágil “construa

projetos em torno de indivíduos motivados. Dê-lhes o ambiente e o suporte que precisam e confie neles para fazerem o trabalho” (FOWLER & HIGHSMITH, 2001), o que faz com que esse resultado seja esperado.

A análise das entrevistas indica que as condições listadas abaixo influenciam a motivação das equipes entrevistadas. Essa relação pode ser vista no quadro 5.

- Senso de Realização dos Membros da Equipe com o Trabalho
- Perspectiva dos Membros da Equipe de Evolução de Carreira dentro da Organização
- Qualidade do Relacionamento entre Membros da Equipe
- Atendimento às Necessidades Básicas de Produção da Equipe
- Constância e Sustentabilidade no Ritmo de Produção da Equipe
- Poder de Decisão e Ação da Equipe quanto à Produção

A última condição é também um dos fatores críticos para a prática de valores Ágeis identificados neste trabalho (veja a seção 4.2.2.6).

As condições que influenciam a motivação dependem fortemente de características individuais dos membros da equipe, de características ambientais e de características culturais. Busca-se nesta seção destacar fatores resultantes da análise das entrevistas que aparentam ser comuns à maioria dos membros de equipes de desenvolvimento quando submetidas a condições semelhantes às dos indivíduos entrevistados.

4.2.2.1.

Senso de Realização dos Membros da Equipe com o Trabalho

O desafio para membros da equipe de trabalhar com tecnologias específicas, difíceis ou pouco conhecidas, que requerem esforço de aprendizado, trabalho de investigação ou especialização, é colocado nas entrevistas como uma condição motivadora no trabalho. Essa condição está de acordo com a Teoria da Fixação de Objetivos (LOCKE, 1996), que afirma que quanto maior a dificuldade de um objetivo, desde que o indivíduo esteja apto a realizá-lo, maior é a realização que ele tem ao cumpri-lo e, portanto, maior é a sua motivação. Pode-se afirmar ainda

que a realização do potencial do indivíduo ao enfrentar desafios técnicos aumenta a sua motivação ao lhe satisfazer necessidades de crescimento, tais como definidas pela Teoria ERC (ALDERFER, 1969; ALDERFER ET AL., 1974). A motivação parece ser ainda maior caso se trate de uma tecnologia nova ou que esteja em destaque no mercado no momento, como se pode observar no seguinte depoimento:

“O que facilita [com relação à motivação] é que é a gente está trabalhando com um pessoal mais novo, que está aprendendo bastante (...) por estar trabalhando com uma tecnologia também que é meio que do momento, (...) uma tecnologia que eles não trabalhavam até então” (Organização 12/Equipe/*ScrumMaster*).

Alguns membros sentem-se desmotivados quando não há desafios. O *ScrumMaster* de uma das equipes, por exemplo, descreve que todos os sistemas desenvolvidos por sua equipe possuem a mesma base de código, de forma que lhes trazem pouquíssimos desafios, o que os desmotiva (Organização 1/Equipe 2/*ScrumMaster*). Outro *ScrumMaster* relata a dificuldade de se equilibrarem trabalhos necessários menos interessantes (como os de suporte ou manutenção), que são atribuídos a sua equipe, com trabalhos mais desafiadores do projeto, que sua equipe divide com uma equipe externa, que fica com a sua maior parte:

“Isso desmotiva qualquer um, não é? Remendar coisa interna (...) enquanto o trabalho legal do projeto, coisa nova, tem que passar para o terceiro. (...) Os caras também querem trabalhar em coisa nova. Ninguém gosta de ficar em manutenção. Então é um dilema psicológico diário, tem que motivar os caras, prometer que vai trabalhar um pouquinho em projeto. Eu passo mais tempo no divã com eles fazendo psicanálise do que gerenciando projeto” (Organização 5/Equipe/*ScrumMaster*).

Equipes que praticam o Scrum devem ser multidisciplinares (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). Para satisfazer essa exigência, de forma geral os membros dessas equipes possuem ou são estimulados a desenvolverem diferentes competências, e isso é visto como um elemento motivador para muitos deles. A Teoria das Características do Trabalho de Hackman et al. (1975) explica esse comportamento, afirmando que a motivação do indivíduo é função do grau em que o trabalho requer que o indivíduo realize uma variedade de atividades diferentes para a sua execução, o que envolve o uso de diferentes conhecimentos e habilidades. Essa questão é relatada por alguns membros entrevistados:

“Aquela coisa de ter vários papéis... isso foi uma das coisas que nos motivou bastante, consegue fazer coisas diferentes” (Organização 4/Equipe/Membro).

“Como nem todos sabiam as mesmas tecnologias, algumas atividades sempre iam para as mesmas pessoas. Com o Scrum, isto diminuiu um pouco. Outras pessoas começaram a aprender as tecnologias menos interessantes. Isto ocorre porque fica claro que existem pessoas fazendo atividades muito legais enquanto outros só fazem as chatas. Alguns acabavam mais motivados que os outros” (Organização 15/Equipe/Membro).

Quando, no entanto, essa multidisciplinaridade é combinada com uma fragmentação no trabalho, de forma que uma mesma tarefa passa pelas mãos dos vários membros da equipe que possuem as habilidades necessárias para realizá-la, isso pode se tornar um fator desmotivador importante, já que não se forma a identidade da tarefa em sua realização (uma das dimensões essenciais, segundo Hackman et al., que provocam a motivação). Assim, o indivíduo não sente que realizou uma parte inteira e identificável do trabalho e não se motiva, como se pode observar no relato de um dos membros de equipe:

“Para mim é muito importante a noção de completude, de pegar alguma coisa e falar ‘eu sou responsável por isso, (...) no final eu vou saber que fiz alguma coisa que faz sentido de alguma forma’. (...) [Em seu projeto, no entanto, acontece diferente:] ‘Eu comecei essa história, mas eu não vou poder terminar ela!’ Porque outra pessoa já pegou metade da história que eu estava fazendo. Então eu não completei nada, não realizei nada, eu não fui responsável, eu não fiz nada desse *software*” (Organização 1/Equipe 1/Membro 1).

Em outro relato sobre o mesmo projeto, o *ScrumMaster* afirma que, embora haja um grande cliente e um contrato, o uso do *software* produzido pela equipe é incerto, já que não há usuários finais definidos. O mesmo acontece em *softwares* produzidos para serem oferecidos ao mercado, sem um cliente certo. De acordo com Hackman et al. (1975), essa incerteza quanto ao uso dos resultados do trabalho afeta a significância da tarefa, ou seja, o impacto positivo que essas tarefas deveriam gerar não é perceptível para os membros da equipe, o que afeta negativamente a sua motivação. Nesse caso específico, a falta de percepção do impacto positivo de seu trabalho é compensada pelos desafios técnicos, o que pode ser observado no seguinte depoimento:

"As pessoas sentem aqui que estão fazendo coisas que ninguém vai usar. É um milagre a equipe estar motivada sem usuário, é só pelo desafio técnico" (Organização 1/Equipe 1/*ScrumMaster*).

A noção de significância da tarefa para membros da equipe também é prejudicada quando a organização não valoriza o trabalho de sua equipe, o que os desmotiva. No relato a seguir, esse problema acontece porque *software* não é o produto final da organização, que tem outra atividade-fim:

"Se você olhar a estrutura funcional da empresa, ela não é uma empresa que o meio dela é tecnologia. O meio dela é cooperativa de saúde. Então para o presidente, a gente muitas vezes é o 'menino do computador'. E não um profissional que tem uma certificação ITIL, Oracle... (...) Numa empresa em que (...) a sua gerência não enxerga você como isso [profissional de TI], (...) às vezes é complicado" (Organização 6/Equipe/*ScrumMaster*).

Mas, de acordo com alguns relatos, nem todos os indivíduos possuem a característica de se motivarem pela realização de desafios técnicos, o que exigiria um perfil específico. Esse comportamento é descrito pela Teoria das Necessidades Adquiridas de McClelland (1961), que afirma que se os membros da equipe possuírem alta necessidade de realização, eles preferirão tarefas desafiadoras em que acreditam que podem obter sucesso, mas se possuírem baixa necessidade de realização, eles buscarão evitar situações de desafio ou utilizarão a dificuldade das tarefas como justificativa para o fracasso. Essa existência dos diferentes perfis é relatada abaixo:

"A equipe é naturalmente motivada. (...) A gente tem desafios técnicos interessantes e uma equipe que é motivada por esta característica. (...) Acabou culminando na troca de pessoas [para atender a este perfil]. (...) Nem todo mundo tem o perfil necessário para este projeto" (Organização 1/Equipe 1/*ScrumMaster*).

Outros entrevistados afirmam que alguns se sentem mais motivados a trabalhar com tecnologias específicas, como se pode notar na seguinte fala de um membro do mesmo projeto:

"O *software* tem vários desafios, a questão da motivação em relação aos desafios, outra questão com relação a... o *software* envolve [uma tecnologia específica] e todo mundo [dessa equipe] gosta [dessa tecnologia específica]" (Organização 1/Equipe 1/Membro 2).

Nota-se que, nesses casos, pode ser necessário retirar do projeto membros da equipe que não possuam o perfil de se motivar por desafios técnicos em geral ou de se motivar com uma tecnologia específica empregada no projeto. Bandura (1977), no entanto, afirma que o nível de dificuldade dos objetivos aceitos pela equipe é função de sua autoeficácia percebida, e essa autoeficácia percebida pode ser aumentada a partir da experiência, treinamento e seleção adequada dos membros da equipe com relação aos conhecimentos e habilidades necessários para o projeto.

4.2.2.2.

Perspectiva dos Membros da Equipe de Evolução de Carreira Dentro da Organização

A análise das entrevistas mostra que, quando há uma perspectiva clara de crescimento profissional para o membro da equipe dentro da organização, sua motivação no trabalho é maior. A busca pela ascensão profissional a partir do reconhecimento do trabalho realizado pelo indivíduo (meritocracia) ou a partir de seu potencial percebido é uma necessidade de crescimento do indivíduo, tal como definida pela Teoria ERC (ALDERFER, 1969; ALDERFER ET AL., 1974), que deve ser satisfeita para motivá-lo. Os depoimentos a seguir de dois indivíduos de uma mesma organização mostram essa questão:

“(...) a questão de você criar um ambiente que faça as pessoas colaborarem, mas ao mesmo tempo terem uma meta, digamos, de carreira, uma coisa que estimule (...) a motivação” (Organização 14/Equipe/*ScrumMaster*).

“A direção da empresa - os sócios e os gerentes - dão uma visão clara para o pessoal dos projetos do que tem sido de importante e tem uma visão clara da meritocracia aqui. Como é uma empresa pequena, não tem aqueles milhares de níveis hierárquicos em que você não enxerga muito bem quais são os critérios para o sucesso. Se tem uma visão clara de que, se você realmente for bem realmente, você vai ser reconhecido e vai chegar mais para a frente, isso te dá uma motivação maior e você tem mais facilidade de ficar comprometido com aquilo” (Organização 14/Equipe/Membro).

Ambos descrevem duas dimensões de crescimento profissional: subir na hierarquia da organização e conseguir trabalhar nos projetos mais importantes para a organização ou nos mais desafiadores. McClelland (1961) afirma, no entanto, que apenas serão motivados pelo crescimento profissional indivíduos com alta necessidade de realização. Já Hackman et al. (1975) afirmam que

somente na presença das dimensões essenciais do trabalho (definidas na Teoria das Características do Trabalho) é que um maior nível de necessidade de crescimento do indivíduo o leva a uma maior motivação para o trabalho.

Quando não há perspectiva de crescimento, os relatos dos entrevistados indicam que a motivação diminui de uma forma geral, como se pode ver abaixo:

“A questão de crescimento interno é realmente difícil, você conseguir melhoria salarial ou ascensão de cargo, isso aqui não existe. Isso é uma coisa externa que vem impactando diretamente na motivação da equipe” (Organização 6/Equipe/*ScrumMaster*).

“O que dificulta é cara do time acomodado. Aquele cara que acha que já chegou num ponto que não vai para lugar nenhum mais. Aí ele dá uma puxada no freio de mão e diz ‘ah, já estou aqui há X anos, já fiz tanto por esse projeto, acho que já está bom’, aí o cara descansa. (...) O cara mais jovem não está no projeto há X anos, ele não está naquela sensação de fim de carreira, ele está querendo crescer” (Organização 3/Equipe/*ScrumMaster*).

Como se pode perceber nas falas dos entrevistados, a falta de perspectiva de crescimento pode acontecer devido a uma política organizacional que não cria condições para o crescimento de seus funcionários, ou pode ser uma questão individual, como quando a pessoa que acredita já ter alcançado o seu máximo em termos de carreira profissional.

4.2.2.3.

Qualidade do Relacionamento entre Membros da Equipe

Um dos aspectos que aparece com frequência nas entrevistas como uma condição que influencia a motivação no trabalho é o relacionamento entre os membros da equipe. Em equipes onde há amizade, confiança e união entre seus membros, como se pode notar nos depoimentos abaixo, há grande motivação:

“[A motivação é facilitada porque] além de ser um ambiente descontraído, todos são amigos” (Organização 15/Equipe/Membro).

“No nosso caso [o que facilita haver motivação] é a amizade que um tem com o outro. Aqui todo mundo é meio família. (...) É um pessoal que interage, conversa, almoça junto, brinca” (Organização 6/Equipe/*ScrumMaster*).

Essas necessidades de relacionamento, de acordo com a Teoria ERC (ALDERFER, 1969; ALDERFER ET AL., 1974), são necessidades básicas do ser

humano no trabalho. Segundo essa teoria, quanto maior o compartilhamento de pensamentos e sentimentos relevantes, maior a motivação do indivíduo no trabalho. Mas, segundo McClelland (1961), indivíduos que se motivam em ambientes onde haja grande interação positiva com outras pessoas e valorizam a equipe cooperativa, complacente com seus membros e coesa possuem alta necessidade de associação, o que os pode levar a sempre buscarem aprovação em suas ações, a evitarem decisões impopulares e a escolherem amigos ao invés de especialistas para trabalharem consigo. Nesses casos, pode haver um aumento na motivação para o trabalho, mas isso pode não ser positivo para os resultados do projeto.

Por outro lado, quando não há um bom relacionamento entre os membros da equipe, os efeitos na motivação dos envolvidos são negativos, podendo levar até mesmo à necessidade de troca de membros da equipe, como é descrito nos relatos abaixo:

“Às vezes uma pessoa não se dá bem com a outra. Isso é uma coisa natural, problemas pessoais. (...) Aí você intercede da forma natural, mostrando que a discordância é uma questão de maturidade, não é uma questão séria. E aquilo se dissolve. Muitas vezes não se dissolve, e aí você dá soluções, passa para outro time. (...) Então essas questões pessoais atrapalham sim” (Organização 14/Equipe/*ScrumMaster*).

“Eu já vi casos de membros da equipe não se darem tão bem e isso desmotivar ambos, e criar um clima chato. E como você precisa de comunicação constante, programação em par e uma série de coisas que você vai ter que lidar com a pessoa diariamente, isso pode gerar um estresse e uma falta de motivação” (Organização 8/Equipe/Membro).

“[Um dos membros da equipe] começou a ter problema, começou a sabotar mesmo a equipe, desmotivar. Ele queria sair. (...) Isso aí foi uma bomba dentro da equipe, muito chato mesmo. Então do ambiente que a gente tinha antes desse projeto para agora, é uma lástima. A gente está tentando reconstruir a partir das cinzas aí” (Organização 5/Equipe/*ScrumMaster*).

Nesse último relato, em particular, mesmo havendo apenas um membro da equipe insatisfeito, as consequências foram extremamente negativas para a motivação de toda a equipe.

4.2.2.4.

Atendimento às Necessidades Básicas de Produção da Equipe

As necessidades básicas para a produção, de acordo com os depoimentos dos entrevistados, incluem um suporte técnico efetivo e uma infraestrutura suficiente (que formam a estrutura básica de apoio à produção, condição para a produção rápida de valor pela equipe, como se pode ver na seção 4.2.4.2), um ambiente de trabalho adequado e salários em dia. Todas essas necessidades, que os entrevistados colocam como motivadoras no trabalho, fazem parte das necessidades de existência do indivíduo, segundo a Teoria ERC (ALDERFER, 1969; ALDERFER ET AL., 1974). Necessidades básicas para a produção aparecem também diretamente ligadas à motivação no princípio Ágil “construa projetos em torno de indivíduos motivados. Dê-lhes o ambiente e o suporte que precisam e confie neles para fazerem o trabalho” (FOWLER & HIGHSMITH, 2001).

Em seguida, pode-se ver um relato em que o suporte técnico ou infraestrutura básica funcionam e outro relato em que não funcionam, afetando diretamente a motivação para o trabalho:

“Quando requisitam compra de equipamento X que facilitaria, então normalmente a gente consegue isso com a liberação de verba com a direção da empresa, e isso mantém as pessoas motivadas. (...) Tudo isso faz parte de manter o indivíduo motivado: você dar uma boa máquina para ele trabalhar, você dar um bom ambiente para ele trabalhar.” (Organização 3/Equipe/*ScrumMaster*).

“O suporte técnico, por exemplo, é uma coisa que deixa a gente meio desconfortável. Questões burocráticas, a gente está tentando conseguir máquina há um tempão, e aí fica cara sem máquina aqui. (...) A motivação poderia ser melhor se a gente tivesse um apoio melhor da empresa” (Organização 1/Equipe 1/Membro 2).

De acordo com os relatos a seguir, dificuldades financeiras da organização, que podem levar a atrasos de salários e à falta de recursos para suprir a infraestrutura básica, influenciam negativamente a motivação para o trabalho:

“O grupo como um todo tem passado por algumas dificuldades financeiras, então a gente não está conseguindo dar a infraestrutura que a gente gostaria para a equipe. A gente tem sentido isso diretamente na motivação. (...) Não adianta eu usar Scrum se eu não consigo ter conexão com a Internet, se eu não tenho uma cadeira confortável para o meu funcionário ou se o salário dele não está em dia” (Organização 10/Equipe/*ScrumMaster*).

“A empresa meio que levou uma baqueada com essa crise agora e teve um problema com a entrada de salário. O que tem interferido na motivação” (Organização 7/Equipe/Membro).

Outro fator que afeta positivamente a motivação é a existência de um ambiente de trabalho que proporcione condições favoráveis ao trabalho de produção, como é descrito nas seguintes falas de entrevistados:

“[O ambiente] é ótimo, excelente. Não tem divisórias, são mesas e nós temos ilhas, cada grupo é uma ilha, as ilhas são vizinhas. Então tudo que cada um precisa todo mundo tenta ajudar na medida do possível” (Organização 9/Equipe/Membro).

“A gente tem um ambiente diferenciado aqui na empresa, (...) a gente tem vários quadros brancos espalhados pela sala de desenvolvimento, a gente tem comida à vontade o dia todo para esses desenvolvedores (...) e a gente preferiu que fazendo o almoço na empresa iria manter a equipe mais unida” (Organização 2/Equipe/*ScrumMaster*).

Percebe-se, pelos relatos, que um ambiente de trabalho com essas características estimula a comunicação e colaboração entre os indivíduos e lhes propicia conforto, motivando-os.

4.2.2.5. Constância e Sustentabilidade no Ritmo de Produção da Equipe

A análise das entrevistas mostra que a produção da equipe de desenvolvimento em um ritmo irregular e eventualmente insustentável é desfavorável para a motivação da equipe no trabalho. A produção realizada em um ritmo constante e sustentável é defendida pelo princípio Ágil “os processos Ágeis promovem o desenvolvimento sustentável. Os patrocinadores, desenvolvedores e usuários devem ser capazes de manter um ritmo constante indefinidamente” (FOWLER & HIGHSMITH, 2001), que estende a sustentabilidade do ritmo a outras partes interessadas no projeto além da equipe, indicando a influência das mesmas nesse trabalho de produção.

As horas extras, que refletem a irregularidade no ritmo de produção e podem levar à sua insustentabilidade, aparecem com frequência nas entrevistas como uma prática corriqueira. Essa prática acaba por afetar profundamente a motivação dos indivíduos quanto ao trabalho, gerando forte rejeição por parte dos

envolvidos na produção. Esse comportamento pode ser percebido na recusa veemente à prática de horas extras de um *ScrumMaster* entrevistado, que tem o apoio organizacional nessa questão:

“Eu não quero fazer hora extra porque tenho vida, entendeu? Como eu não quero fazer, não quero que ninguém faça também. Aqui a gente não faz hora extra, não trabalha sábado, não faz nada disso. É uma questão de filosofia da empresa” (Organização 1/Equipe 2/*ScrumMaster*).

A aceleração exagerada do ritmo de produção e o uso intensivo de horas extras em momentos considerados críticos do projeto - um exemplo frequente disso é a perspectiva de atraso em uma entrega prevista - provoca, como se pode observar no depoimento abaixo, um descontentamento e uma consequente diminuição na motivação da equipe quanto ao trabalho:

“Ninguém gosta de trabalhar até de madrugada todo dia, nem de perder feriado, nem de abdicar de sua vida pessoal. Isso obviamente começou a fazer a motivação cair. Ninguém aguentava mais o projeto, começou a rezar para o projeto acabar” (Organização 13/Equipe/Membro).

Esse tipo de solução gera os desperdícios *muri* (sobrecarga nas pessoas ou equipamentos além dos limites naturais) e *mura*, (irregularidades, flutuações ou desbalanços no ritmo de produção), conforme definidos por Hampson (1999) na Produção Enxuta, o que afeta negativamente o ritmo da produção. Segundo esse autor, esses desperdícios devem ser evitados utilizando o *heijunka*, ou balanceamento e nivelamento da produção, com ritmos e níveis de trabalho constantes e sustentáveis. A dificuldade frequente de equipe em evitá-los, gerando sobrecarga de trabalho, pode ter forte efeito desmotivador sobre seus membros.

4.2.2.6.

Poder de Decisão e Ação da Equipe Quanto à Produção

O poder de decisão e ação da equipe quanto à produção, além de se destacar nas entrevistas como um fator crítico para a prática de valores Ágeis (veja seção 4.2.6), aparece também como condição que influencia a motivação dos membros da equipe no trabalho. Assim, quanto maior esse poder de decisão da equipe, maior a motivação de seus membros. Essa afirmação está de acordo com a afirmação de Cohen (1993), na qual é razoável se esperar que a participação em

equipes autogerenciadas, que possuem maior poder de decisão e ação (KOZLOWSKI & BELL, 2003; HACKMAN, 1987; MOSCOVICI, 1996), cause um impacto positivo na satisfação dos membros da equipe quanto a seu trabalho, equipe, relações sociais e oportunidades de crescimento. Esse impacto positivo pode ser interpretado como um aumento na satisfação das necessidades básicas de existência, de relacionamento e de crescimento do homem que, segundo Alderfer (1969) e Alderfer et al. (1974), é determinante para a motivação do indivíduo no trabalho. O aumento na motivação decorrente do maior poder de ação e decisão é ilustrado nos depoimentos a seguir:

“Nosso time tem muito mais responsabilidade com o cliente, e sentem realmente que eles são os principais responsáveis pelo projeto. Então, nesse aspecto, a equipe tem tido uma motivação muito interessante para assumir mais responsabilidades do projeto” (Organização 2/Equipe/*ScrumMaster*).

“(…) já ocorreu de inúmeros profissionais serem convidados por concorrentes e eles não saem nem para ganhar mais, (…) porque os desenvolvedores passaram a ter voz, passaram a ser decisivos (…) na qualidade daquele produto. (…) E a questão da pressão ser disseminada no time inteiro, não ser focada em cima de uma pessoa. E a comemoração de uma vitória dentro de um ciclo. Então são uma série de ferramentzinhas que fazem o ambiente saudável. Isso vem em cima desses princípios do time ter poder de decisão, poder de ação dentro de um projeto” (Organização 11/Equipe/*ScrumMaster*).

“A forma que tu trata o colega [motiva]. Num sistema mais tradicional, tem o analista, que é o detentor do conhecimento que vai dizer o que tem que se fazer ou não. Então aqui é diferente, a gente é colega e todos vão se ajudar, todos vão compartilhar o conhecimento. Então não tem esse nível hierárquico. (…) o pessoal se sente mais à vontade para propor” (Organização 4/Equipe/*ScrumMaster*).

Esse maior poder de decisão e ação pode se traduzir também em uma maior flexibilidade na definição dos horários de trabalho dos membros da equipe, o que aumenta a sua motivação, segundo os relatos abaixo:

“Uma coisa que eu gosto muito é a liberdade de tempo que eu tenho. Evento, treinamento, consultoria, todas essas coisas, muitas vezes eu tive que me afastar do projeto por meio dia, durante um dia, durante alguns dias da semana. Só que se não tivesse esse cenário, para mim eu me sentiria perdendo várias coisas” (Organização 8/Equipe/Membro).

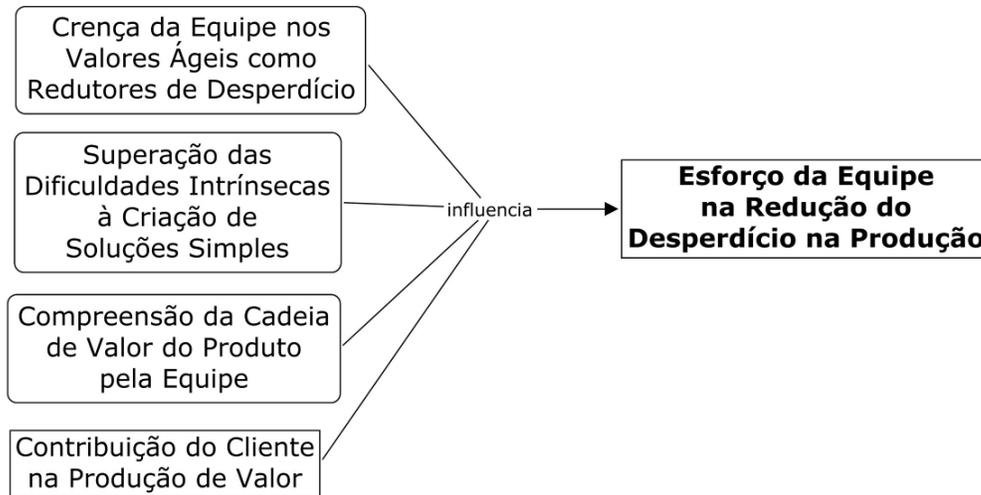
“Para a pessoa estar focada e motivada no *sprint* ela tem que estar feliz com a família. Então ele não vai conseguir produzir direito no *sprint* se a esposa dele, por exemplo, estiver doente e ele precisa levar ela ao médico e ele não tem a liberdade para sair da cadeira para levar ela ao médico” (Organização 3/Equipe/*ScrumMaster*).

“A gente tem a questão de usar a sexta-feira para que a equipe desenvolva projetos pessoais, então eles trabalham no *sprint* apenas de segunda a quinta. Sexta-feira a gente tem o desenvolvimento de produtos criados aqui pelos desenvolvedores e isso motiva bastante a equipe” (Organização 2/Equipe/*ScrumMaster*).

Esses relatos indicam que, sem comprometer a responsabilidade e o compromisso com os objetivos organizacionais, os membros da equipe desejam ter liberdade de, durante o horário de trabalho, atender a emergências familiares, participar de eventos e treinamentos e até mesmo de dar andamento a projetos pessoais (que eventualmente pertencerão à empresa), o que afeta sua motivação para o trabalho.

4.2.3.

Esforço da Equipe na Redução do Desperdício na Produção



Quadro 6: Fator crítico “esforço da equipe na redução do desperdício na produção” e condições que influenciam sua manifestação

A redução do desperdício na produção de *software* permeia praticamente todos os valores e princípios Ágeis, assim como é uma das principais preocupações da Produção Enxuta.

A Produção Enxuta defende que o desperdício no trabalho (*muda*) deve ser eliminado (WOMACK & JONES, 1998). Outros dois tipos de desperdício, o *muri* (sobrecarga nas pessoas ou equipamentos além dos limites naturais) e o *mura* (irregularidades, flutuações ou desbalanços no ritmo de produção) também devem ser eliminados nos processos de produção (HAMPSON, 1999).

Nas práticas Ágeis, a busca pela maximização da comunicação dentro da equipe e entre os membros da equipe e o cliente, através das entregas frequentes e das reuniões programadas, pretende propiciar inspeções frequentes do processo e do produto. Assim, através do *feedback* obtido por essas inspeções, pretende permitir a adaptação do processo e do produto através de melhorias, que dessa forma reduzem o desperdício (LARMAN, 2003; HIGHSMITH, 2004). Outra questão é a valorização dos indivíduos e de suas interações mais que processos e ferramentas, pois se observa que o maior foco nos dois últimos, ou seja, no uso excessivo de ferramentas e de processos complexos, ao invés de nos dois primeiros, gera desperdícios (LARMAN, 2003).

Em termos dos princípios Ágeis (FOWLER & HIGHSMITH, 2001), o princípio “simplicidade - a arte de se maximizar a quantidade de trabalho não feito - é essencial”, defende a redução do desperdício através da simplicidade, ou seja, a geração de somente o que é realmente necessário e suficiente como produto do trabalho, passando somente por etapas necessárias e da forma mais simples possível. Percebe-se pelas entrevistas que é muito comum em projetos de *software* a geração de funcionalidades mais complexas e abrangentes do que o necessário, algo combatido através da prática dos valores Ágeis.

Já o princípio Ágil “*software* em funcionamento é a medida primária de progresso” busca eliminar o desperdício criado pelo uso de outras métricas do progresso do desenvolvimento do produto que não a entrega de valor propriamente dito. Esse princípio é uma resposta a uma das grandes fontes de desperdício em projetos, que é a geração de funcionalidades ou artefatos (como documentos, planos e relatórios) que não criam qualquer valor para o cliente e, na maioria dos casos, sequer foram solicitados por ele.

Pode-se afirmar também que o princípio Ágil “a atenção contínua à excelência técnica e a um bom projeto aumenta a agilidade” pretende reduzir o desperdício buscando evitar produtos mal projetados e realizados por indivíduos sem o conhecimento e habilidades necessárias.

Liker (2003) define sete tipos de *muda* combatidos na Produção Enxuta, muitos dos quais são semelhantes aos combatidos pelos valores e princípios Ágeis na produção de *software* como, por exemplo, o excesso de produção (geração de funcionalidades no *software* mais complexas e abrangentes do que o necessário), o estoque (geração de funcionalidades ou artefatos não solicitados), o processamento desnecessário (uso de etapas desnecessárias na produção das funcionalidades, decorrente da falta de inspeção e adaptação no processo) ou incorreto (produção de *software* sem excelência técnica e mal planejada) e o projeto de produtos e serviços que não atendem às necessidades do consumidor (falta de inspeção e adaptação no produto, decorrente da má comunicação com o cliente).

A análise das entrevistas indica que as condições a seguir influenciam o esforço que a equipe empenha na redução do desperdício na produção de *software*. O quadro 6 sintetiza a relação entre essas condições e o fator crítico.

- Crença da Equipe nos Valores Ágeis como Redutores de Desperdício
- Superação das Dificuldades Intrínsecas à Criação de Soluções Simples
- Compreensão da Cadeia de Valor do Produto pela Equipe
- Contribuição do Cliente na Produção de Valor

A última condição é também um dos fatores críticos para a prática de valores Ágeis identificados neste trabalho (veja a seção 4.2.3.4).

4.2.3.1.

Crença da Equipe nos Valores Ágeis como Redutores de Desperdício

A equipe acreditar ou não que a prática de determinados valores e princípios Ágeis ajuda a reduzir o desperdício aparece com frequência nas entrevistas como a mais importante condição influenciadora no empenho da equipe na diminuição do desperdício em seus processos de produção. Essa crença muitas vezes vem com a experiência, conforme mostra o depoimento a seguir, em que a equipe entendeu o valor da simplicidade ao ter problemas por não praticá-la:

“As experiências que a gente teve no passado de não ter feito da maneira mais simples hoje causam problemas para a gente na manutenção. (...) Então a equipe aprende com esses erros quando encontra isso no desenvolvimento dos itens” (Organização 2/Equipe/*ScrumMaster*).

Em outros casos, a prática da simplicidade já faz parte da cultura da empresa e da equipe, o que faz com que a equipe mantenha uma preocupação constante com a redução de desperdícios a partir desse princípio, como se pode ver nos relatos a seguir:

“Sempre que a gente pensa: ‘ah, vamos fazer tal coisa, tal coisa’, tem sempre dois ou três para falar: ‘gente, precisa mesmo disso? Isso aí vai agregar o quê? Vai gastar quanto tempo para fazer?’ (...) Hoje eu posso falar que isso é uma cultura disseminada” (Organização 8/Equipe/Membro).

“Tendo gente chata toda hora reclamando quando alguém faz uma solução complexa e refatorando as coisas para ficarem mais simples, e tem até coisas coladas pela parede dizendo que simplicidade é bom e complexidade é ruim” (Organização 1/Equipe 1/*ScrumMaster*).

“O que é o simples nesse projeto? A gente sempre usa essa palavra: simples, simples, simples. O que é o simples? (...) Vamos descomplicar, o que é o simples? Funciona que é uma beleza. (...) Isso, como te falei, como a empresa comprou a ideia, isso está dentro...” (Organização 14/Equipe/*ScrumMaster*).

Dessa forma, a própria equipe busca a simplicidade através da auto-observação, autoavaliação e autorregulação, o que é possível devido à existência na equipe de normas comportamentais intensas e cristalizadas que estimulem esse comportamento, de acordo com Hackman (1987), e que pode, além disso, ser facilitado pelo estímulo de um líder, conforme descrito por Manz & Sims (1987).

No relato seguinte, o foco maior na produção de *software* de valor do que na produção de documentação ou modelos faz parte da cultura da organização, o que também faz com que a equipe busque a redução de desperdícios na produção:

“Isso hoje está bem aceito, é a cultura mesmo da empresa. A gente não fica com excesso de documentação ou de modelos. É uma coisa que é bem focada no *software*. Não que não exista documentação, obviamente, mas hoje a empresa realmente abraçou isso aí, esse lado” (Organização 10/Equipe/*ScrumMaster*).

Existem, no entanto, duas situações muito comuns em que as equipes evitam a simplicidade e a produção focada em valor para o cliente, buscando soluções complexas ou produzindo funcionalidades e artefatos não requisitados e, assim, acabam por gerar desperdícios. A primeira deriva de uma crença em que, ao buscar a complexidade ou antever funcionalidades, a equipe pode se antecipar a problemas e a aquisições futuras ou evitar retrabalho. Seguindo essa linha de pensamento, os entrevistados abaixo não acreditam na simplicidade como redutora de desperdícios:

“Eu acho que o excesso de simplicidade vai prejudicar a sua qualidade, manutenção e extensão. (...) [No projeto,] a gente complicou um pouco mais para nos facilitar no futuro, (...) a simplicidade vai acontecer por estar fácil de manter o código. (...) [Para saber o que vão precisar no futuro] é meio que um *feeling* [sentimento] nosso e a gente tem acertado com frequência” (Organização 4/Equipe/*ScrumMaster*).

“Nem sempre tem uma discussão para saber se ‘olha, será que é realmente bom que isso seja simples? Será que a gente já não poderia fazer uma coisa mais geral?’ (...) Se fizesse de uma maneira, ia facilitar no futuro, para uma determinada coisa que talvez a gente fosse fazer, mas algumas vezes as pessoas não concordam comigo. Então fazem da forma mais simples” (Organização 1/Equipe 1/Membro 1).

Assim, indivíduos que defendem os valores Ágeis na organização encontram barreiras tanto em suas equipes quanto em outros indivíduos ou departamentos dos quais suas equipes dependem no processo de produção, como acontece nos relatos a seguir:

“Foi até difícil convencer o pessoal de qualidade de abrir mão de algumas coisas. Mas aí, o nosso líder técnico, ele tem muita força aqui dentro e ele conseguiu. (...) Claro, sempre tinha alguém de qualidade acompanhando, era uma negociação constante, algumas coisas eles não abriam mão. (...) Não conseguiu abrir mão totalmente, não conseguiu anular. Mas tornou isso mais leve” (Organização 13/Equipe/Membro).

“Acho que é a busca de fazer aquele *design* perfeito, que vá funcionar em qualquer situação, que já vá prever o que a gente vai fazer, não vá ter trabalho no futuro, mas eu acho isso igual a jogar na loteria. (...) Então tenho batido um pouco com a equipe com relação a isso e está sendo difícil convencer do contrário” (Organização 10/Equipe/*ScrumMaster*).

A segunda situação em que as equipes não praticam a simplicidade e acabam por gerar desperdício ocorre quando a produção de documentação excessivamente detalhada faz parte da cultura da organização, o que geralmente deriva da crença de que, ao não documentar, haverá perda de informações que poderiam ser usadas no futuro, como se pode ver no depoimento abaixo:

“Qualquer coisa que a gente faça a gente documenta. (...) Às vezes é um pouco mais burocrática, às vezes é um pouco mais pelo medo de precisar daquela informação um dia e não ter. Às vezes nunca mais vai usar aquilo, mas a gente documenta. Que eu saiba, 30-40% da documentação a gente usa. (...) Acho também que é perfeccionismo demais, tentar deixar as coisas certinhas demais, documentadas demais (...) O time tem uma cultura de documentar, a gente se acostumou a documentar as coisas. A gente quer manter a casa organizada” (Organização 1/Equipe 1/Membro 2).

A produção de documentação detalhada também faz parte da cultura da organização quando existe a crença de que a documentação é necessária para proteger a organização ou a equipe de cobranças do cliente, se houver algum tipo de contestação quanto ao que foi combinado ou contratado. O relato a seguir mostra um caso em que isso ocorre:

“[O gerente] ainda tem algumas coisas arraigadas do processo padrão de requisitos, documentação, abrir termo. (...) Um exemplo típico. Um colega da gente orçou um requisito em 24 horas, que seriam quatro dias de trabalho. Só que, quando ele começou a mexer no código, ele viu uma alternativa que ele resolveu em 6 horas.

Só que está de fora do projeto, quem é focado em requisito foi questionar ele por que ele fez errado. (...) E aí foi a discussão na sala eu defendendo essa perspectiva da metodologia Ágil e o pessoal achando que não, que tem que ter requisito, que tem que ter papel, tem que ter isso, tem que ter aquilo” (Organização 6/Equipe/*ScrumMaster*).

Como se pode perceber, esse comportamento deriva da persistência na organização de culturas tradicionais de gerência de projeto contrárias aos valores Ágeis.

4.2.3.2. Superação das Dificuldades Intrínsecas à Criação de Soluções Simples

Entrevistados relatam que criar soluções simples, de forma a evitar o desperdício, é muitas vezes mais difícil do que criar soluções complexas. Essa dificuldade acontece porque a solução simples que funciona e atende aos requisitos é, em geral, menos óbvia e menos visível do que uma solução mais complexa, porém de rápida definição. Essa questão é citada por um dos entrevistados, que afirma que “a solução simples às vezes não é óbvia, tem que pensar um pouco em metanível assim, refletir um pouco” (Organização 1/Equipe 1/*ScrumMaster*). Por essa razão, muitas vezes o desenvolvedor de *software* prefere criar soluções mais complexas, mesmo que isso signifique a geração de desperdício. Em outros casos, o indivíduo tem dificuldade em visualizar alternativas ao que ele já está fazendo, e uma ajuda externa de um líder ou de um colega pode fazê-lo enxergar uma solução mais simples, como mostra o depoimento a seguir:

“[Uma desenvolvedora estava com dificuldades com um desafio técnico] Eu percebi que ela estava travada ali num estado mental que ela não conseguia enxergar outra solução. E estava complicando cada vez mais. É o momento de você tirar a pessoa do estado mental onde ela está, abrir a caixa, mostrar a ela outras oportunidades e assim, conseguir trazer de volta a simplicidade. Em algoritmos também” (Organização 5/Equipe/*ScrumMaster*).

Um problema frequente é a confusão de desenvolvedores entre realizar o trabalho com simplicidade e realizar o trabalho sem qualidade, sem boas práticas ou sem o devido planejamento. A palavra “gambiarra” é utilizada para descrever essa situação no depoimento abaixo:

“O foco era resolver. Mesmo que resolver o problema desse um produto de má qualidade. (...) Às vezes a simplicidade era confundida com gambiarra. Hoje a gente tem focado no conceito mesmo de simplicidade e batido nesse conceito de gambiarra, tentando acabar com essa história de gambiarra” (Organização 6/Equipe/*ScrumMaster*).

Em outros casos, a equipe pode usar a simplicidade como desculpa para gerar soluções rapidamente sem projetá-las adequadamente e sem levar em consideração uma visão do produto. Percebe-se pelas entrevistas que não é uma tarefa fácil distinguir o quanto uma solução deve ser projetada tendo em vista possíveis usos e consequências futuras, e o quanto disso é desperdício. Um dos *ScrumMasters* descreve uma situação em que algo semelhante ocorre em sua equipe:

“Tem uma série de validações que tem que fazer numa parte ali e o pessoal estava fazendo com simplicidade, fazendo só o que precisava ali. E não estava vendo que, no futuro, essas validações vão ser reutilizadas em outras partes do sistema. (...) Muitas vezes o pessoal confunde agilidade em resolver rápido e resolver daquela forma para entregar rápido e não se preocupar com a manutenção” (Organização 4/Equipe/*ScrumMaster*).

Contrário à prática da “gambiarra” e do mau projeto, o princípio Ágil “a atenção contínua à excelência técnica e a um bom projeto aumenta a agilidade” (FOWLER & HIGHSMITH, 2001) indica que qualquer que seja a solução adotada, ela deve ser bem projetada e executada com excelência técnica.

Em projetos mais antigos que possuem código legado, ou seja, funcionalidades produzidas por outra equipe ou outras equipes ao longo do tempo, a simplicidade se apresenta mais difícil de ser realizada, de forma geral. Os relatos abaixo descrevem alguns casos em que isso ocorre:

“O que atrapalha é o dia-a-dia. É você ter que lidar com código legado. Ter que lidar com projeto legado que está todo mal modelado. Não está nada simples. Como é que é o simples desse negócio? Até onde posso mexer, sem impactar o negócio? De que forma eu cresço de forma orgânica?” (Organização 14/Equipe/*ScrumMaster*).

“Quando a gente pegou o projeto, era um projeto (...) [que] tem um código bem antigo, que passou por diversas empresas, por diversas pessoas. Então a gente não tinha um padrão de código. (...) Nosso problema mesmo é com coisas que já estão lá há bastante tempo, que a gente tem que mexer, a gente não tem aquele conhecimento do código” (Organização 4/Equipe/Membro).

Essa dificuldade ocorre devido ao desconhecimento do código, que não foi produzido por essa equipe que hoje lhe dá manutenção e adiciona novas funcionalidades. Percebe-se que nesses casos a simplicidade se apresenta ainda mais difícil de ser realizada quando não foram utilizados padrões bem definidos de código e boas práticas de engenharia de *software* na geração desse código legado, o que parece ser o caso mais comum.

Outra dificuldade intrínseca à criação de soluções simples decorre do orgulho que muitos programadores sentem ao criar soluções complexas ou ao criar soluções que acreditam que surpreenderão os clientes, como é ilustrado nas entrevistas a seguir:

“Tem uma questão de florear, de construir coisas que não são o fim do negócio. Não são nem o meio, nem o fim. São coisas que estão ali para dar uma sensação maior de, às vezes, conforto, e tal. É a tal da perfumaria. (...) O desenvolvedor gosta muito de mostrar o que o cliente não vai ver. Ele fala: ‘nossa, fiz um código aqui que dá um triplo mortal carpado para trás’. No final das contas a funcionalidade é a mesma. É um orgulho, assim, não é? Orgulho e ego são coisas extremamente difíceis” (Organização 8/Equipe/Membro).

“Eu acho que o alter-ego técnico de pessoas que são altamente técnicas, ela dificulta a simplicidade. Um exemplo bem mais fácil, bem mais real para ti, que eu te digo, é a linguagem. O cara bota ‘*log in*, usuário’. Aí ele bota o seguinte: ‘olha, aconteceu um problema em sua *query*’. Imaginar que um usuário, operário, que trabalha em caixa de loja vai entender expressões como *log in*, como *query*” (Organização 9/Equipe/Membro).

“O programador sempre quer fazer alguma coisa a mais, além do que o cliente pediu. (...) Sempre quer entregar alguma coisa para surpreender o cliente, uma coisa a mais, e a gente está sempre falando que, se o cliente pediu aquilo, aquilo é a qualidade para o cliente, a qualidade está nos olhos do cliente. Então a gente fazer a mais ou fazer a menos, não é fazer aquilo que o cliente está pedindo” (Organização 2/Equipe/ScrumMaster).

Essa tendência a complicar o que poderia ser feito de forma mais simples parece ser uma característica natural de desenvolvedores que valorizam muito a parte técnica de seu trabalho, que por isso acabam por focar nessa dimensão em detrimento do valor de negócio.

4.2.3.3.

Compreensão da Cadeia de Valor do Produto pela Equipe

No contexto de desenvolvimento Ágil de *software*, podemos definir a cadeia de valor, a partir da definição de Womack & Jones (1998) na Produção Enxuta,

como todas as atividades específicas realizadas para que se possa oferecer ao cliente um incremento pronto no produto. Entre essas atividades, aquelas que não agregam valor ao produto (e que, portanto, são desnecessárias) são desperdício e devem ser eliminadas imediatamente (*muda* Tipo Dois, de acordo com Womack & Jones, 1998). No entanto, aquelas que não agregam valor ao produto mas são necessárias ou inevitáveis também são desperdício, mas não podem ser eliminadas, ao menos naquele momento (*muda* Tipo Um).

No desenvolvimento de *software*, deve-se diferenciar, por exemplo, o caso em que a documentação extensa constitui um desperdício dispensável do caso em que a documentação extensa é uma exigência externa, necessária para a própria viabilização do projeto. Pode-se interpretar que essa grande documentação, nesse último caso, constitui um desperdício que não pode ser eliminado, conforme definido por Womack & Jones (1998). Casos comuns em que há exigências de documentação extensa incluem projetos em que o cliente é uma grande empresa, um órgão governamental ou uma instituição governamental, nos quais há geralmente muita burocracia envolvida, como pode ser visto no depoimento a seguir:

“A gente teve um caso que a gente precisou fazer um projeto específico para a (...) Prefeitura aqui de [cidade onde fica a empresa]. E a gente percebeu que, para esse tipo de projeto, a gente precisava ter um nível maior de documentação, de formalismo. (...) Se o cliente está exigindo que precisa ter uma documentação maior, vai ter que fazer. (...) Eu vejo como uma certa insegurança, ou até poderia ser uma falta de confiança entre fornecedor e cliente. Todos os projetos que já vêm do passado, de outros fornecedores, das histórias que a gente escuta, não é? Então você já entra numa relação com o cliente novo com uma bagagem que não foi culpa sua, da sua empresa” (Organização 10/Equipe/*ScrumMaster*).

Um erro comum no desenvolvimento Ágil de *software* é a crença de que toda e qualquer documentação é um desperdício dispensável. Segundo Cockburn (2007), deve-se produzir somente a documentação necessária e suficiente para a realização do trabalho, o que depende de fatores intrínsecos ao projeto ou ao produto a ser desenvolvido. No relato abaixo, percebem-se diversos problemas decorrentes de não se gerar uma documentação mínima necessária específica para o projeto:

“A gente não tem feito muita documentação. Isso traz problemas porque a gente, em não documentando, tem problemas inclusive no suporte de segundo nível.

Então a gente não está documentando e daí chega a regra e está na cabeça de alguém, está escrito numa *issue* [item] do [sistema de controle de erros], está em algum lugar perdido, não está numa documentação, num local de fácil acesso a todos. Então esse é um problema sério que a gente tem” (Organização 4/Equipe/*ScrumMaster*).

Para projetos muito grandes, com regras de negócio complexas, pode ser necessário um grande nível de documentação e de ferramentas. Em seguida, pode-se ver um depoimento sobre um projeto muito complexo que aparentemente exige ferramentas e processos complexos para facilitar seu desenvolvimento:

“A gente gera o *backlog* e também faz uma atualização de modelo UML. (...) A gente pega esse modelo e utiliza um gerador de classes que (...) gera todos os tipos que a gente vai precisar naquela fase. E a gente também tem um outro artefato que a gente construiu aqui que serve de auxílio durante a refatoração do projeto. Quando a gente começa a fase, a gente já tem todo um corpo de classes, de código, que a gente não precisa digitar. (...) O processo bem definido de quando ocorre cada coisa, desse facilitador que remove trabalho braçal e diminui também a quantidade de erros (...). As nossas entregas de duas semanas tem sido bem pouco traumáticas por conta disso” (Organização 8/Equipe/Membro).

A análise das entrevistas, no entanto, mostra que é difícil a tarefa de se distinguir o que é e o que não é desperdício nesses casos, pois essa distinção depende da interpretação de cada equipe sobre como o trabalho deve ser executado e sobre o que é ou não necessário para tal.

4.2.3.4. Contribuição do Cliente na Produção de Valor

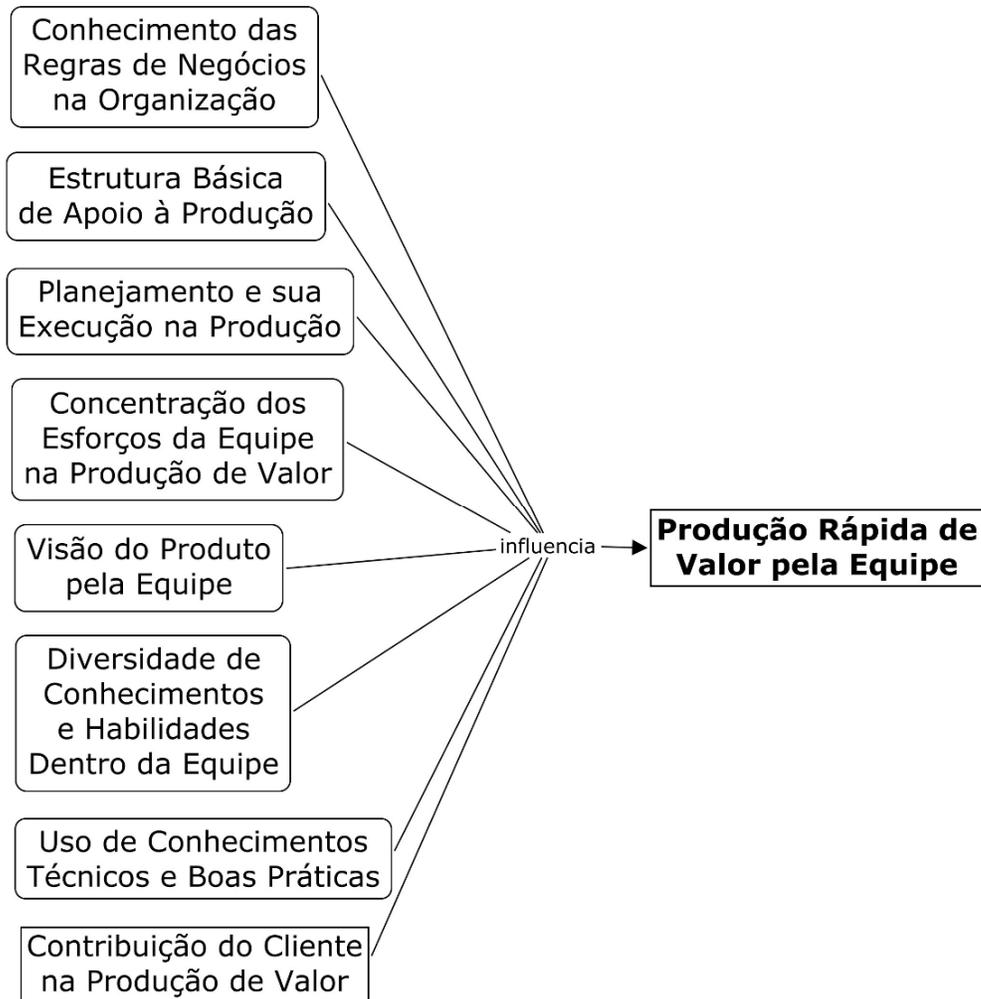
A contribuição do cliente na produção de valor, um dos fatores críticos para a prática de valores Ágeis, foi identificada a partir da análise das entrevistas como uma condição que influencia o esforço da equipe na redução do desperdício.

Destacam-se algumas situações em que a influência dessa condição parece manifestar-se com maior intensidade. Na primeira, a equipe gera desperdício quando, por qualquer razão, o cliente não colabora efetivamente para a definição dos requisitos que devem ser implementados pela equipe em cada ciclo de desenvolvimento, já que a equipe gastará esforço criando funcionalidades que têm menor chance de serem usadas pelo cliente. Em outra situação, quando há interferências do cliente no processo de produção da equipe (solicitando suporte, por exemplo, ou a implementação novos requisitos urgentes não planejados),

também há geração de desperdício pela equipe, já que essas interferências atrapalham o planejamento e a produção das funcionalidades de maior valor.

Os impactos desse fator na prática dos valores Ágeis serão explicados com maiores detalhes na seção 4.2.7, que aborda essa questão.

4.2.4. Produção Rápida de Valor pela Equipe



Quadro 7: Fator crítico “produção rápida de valor pela equipe” e condições que influenciam sua manifestação

A produção rápida de valor para o cliente pela equipe de desenvolvimento de *software* emerge das entrevistas como um fator crítico para a prática de valores Ágeis. Esse é naturalmente um resultado esperado, visto que a produção rápida de valor é necessária para a entrega frequente de valor para o cliente, defendida pelos princípios Ágeis “nossa maior prioridade é satisfazer o cliente através da entrega contínua e desde cedo de *software* com valor” e “entregar frequentemente *software* em funcionamento, desde a cada duas semanas até a cada dois meses, com uma preferência por prazos mais curtos” (FOWLER & HIGHSMITH, 2001).

A Produção Enxuta, de acordo com Womack & Jones (1998), define que os passos necessários para a geração de valor devem fluir continuamente, um após o outro, sem paradas, lotes, filas ou fronteiras entre departamentos de forma que o produto seja produzido e entregue quando o cliente necessita, em uma produção “puxada”. De forma semelhante, no desenvolvimento Ágil de *software* com Scrum, incrementos no produto de maior valor para o cliente são gerados por equipes multidisciplinares em um fluxo contínuo e sem interrupções, dentro de iterações curtas, que objetivam entregas frequentes (LARMAN, 2003).

A partir da análise das entrevistas, foi detectado que as condições abaixo exercem influência na produção rápida de valor pela equipe. O quadro 7 mostra essa relação.

- Conhecimento das Regras de Negócios na Organização
- Estrutura Básica de Apoio à Produção
- Planejamento e sua Execução na Produção
- Concentração dos Esforços da Equipe na Produção de Valor
- Visão do Produto pela Equipe
- Diversidade de Conhecimentos e Habilidades Dentro da Equipe
- Uso de Conhecimentos Técnicos e Boas Práticas
- Contribuição do Cliente na Produção de Valor

A última condição é também um dos fatores críticos para a prática de valores Ágeis identificado neste trabalho (veja a seção 4.2.4.8).

4.2.4.1.

Conhecimento das Regras de Negócios na Organização

A existência, dentro da própria organização, de especialistas ou conhecedores das regras de negócios necessárias para o desenvolvimento do produto aparece frequentemente como uma condição que influencia a produção rápida de valor pela equipe. Por exemplo, para o desenvolvimento de um sistema fiscal, deve haver especialistas em contabilidade próximos à equipe de desenvolvimento para sanar quaisquer dúvidas sobre como o produto deve ser

feito. A análise das entrevistas mostra que, quanto mais próximos esses especialistas estiverem da equipe, melhor e mais rapidamente essa equipe produzirá valor para o cliente.

O princípio Ágil “pessoas de negócio e desenvolvedores devem trabalhar em conjunto diariamente por todo o projeto” (FOWLER & HIGHSMITH, 2001) já estabelece a necessidade da proximidade entre a equipe de desenvolvimento e os conhecedores das regras de negócios.

Muitas das equipes entrevistadas possuem alguns ou todos os seus próprios desenvolvedores capacitados nas regras de negócios do produto. Essa prática se mostra bastante efetiva, já que aqueles que realizam o trabalho entendem as necessidades dos usuários do produto final. Os relatos a seguir ilustram essa questão:

“Nós temos os especialistas, por exemplo, eu sou muito direcionada para a parte fiscal. (...) Tem um colega meu que é na parte de nota fiscal eletrônica, entendeu? Então a gente tem uma interação constante com o suporte, com quem lida diretamente com o cliente, muitas vezes até com o cliente, no caso de não haver recurso. Então estamos por dentro do que é feito, do que realmente o cliente necessita” (Organização 3/Equipe/Membro).

“A nossa equipe faz o levantamento do negócio, a nossa equipe ajuda o P. O. a definir os processos de negócio da empresa dele. (...) Então basicamente, a própria equipe de desenvolvimento também atua como analistas de negócio, e o P. O. só aprova as definições junto com os seus *stakeholders*. A equipe tem um conhecimento do negócio inteiro, muito grande” (Organização 2/Equipe/*ScrumMaster*).

“O analista de negócios não conhece a parte de desenvolvimento... não conhece a fundo, ele só tem uma noção. Mas o desenvolvedor aqui ele conhece o negócio, então eles se complementam. O desenvolvedor aqui não é codificador. (...) Tanto já vêm desenvolvedores especializados do mercado, quanto a gente treina aqui. Aqui tem programas de estágio, e por ter programa de estágio a gente acha muito interessante ver um rapaz de vinte anos que chega aqui só sabendo lógica e SQL [linguagem de banco de dados] etc., e ele pega essa oportunidade e se desenvolve. A gente dá essa oportunidade” (Organização 9/Equipe/Membro).

Outro caso comum acontece quando há especialistas de negócios dentro da organização, mas fora da equipe de desenvolvimento. Esses especialistas podem ser consultores que atuam especificamente no projeto (muitas vezes até mesmo como *Product Owners*) ou um setor da organização responsável por fornecer informações aos desenvolvedores quanto às regras de negócios. Ambos os casos podem ser vistos a seguir:

“O ‘Produto’, que seria o pessoal do negócio, eles estão em um andar acima. Então a gente tem bastante contato com eles. A gente faz reuniões diárias com eles (...) e a gente tem também contato aqui por *e-mail*, por MSN, telefone. Sempre que a gente tem uma dúvida, a gente entra em contato com eles. A comunicação entre nós é boa” (Organização 4/Equipe/Membro).

“A gente tem junto da equipe aqui o dia todo um analista de negócios. Ele trabalha para a empresa, (...) é o P. O. do projeto. Vamos dizer que ele fica 80% do tempo junto com a equipe. (...) Então enquanto o *sprint* está correndo, ele fica disponível para tirar dúvidas da equipe” (Organização 8/Equipe/Membro).

“A partir de um determinado ciclo X, a gente começou a colocar um analista de negócios desde a reunião de planejamento. (...) O analista de negócios, que fazia parte mais do time periférico do P. O., ele estava sempre disponível por causa dessa necessidade de interação das pessoas. De tirar dúvida mesmo, de estar ali, de já validar” (Organização 11/Equipe/*ScrumMaster*).

Em outros casos, no entanto, os especialistas nas regras de negócios não estão disponíveis para a equipe de desenvolvimento, como se pode ver nos relatos abaixo.

“A gente não tem muita ligação com as pessoas de negócios. (...) Elas estão no cliente e nem sequer estão disponíveis, mesmo no cliente” (Organização 1/Equipe 1/*ScrumMaster*).

“O que dificulta é que às vezes quando esse cara [analista de negócios] está fora do time e está em outro departamento da empresa, é ter a disponibilidade dele para a gente. Porque muitas vezes eu preciso da opinião dele. Para continuar a coisa eu preciso que ele interfira e dite as regras. Ele precisar estar lá para dizer como a coisa deve acontecer e o cara não vem. E isso atrapalha” (Organização 3/Equipe/*ScrumMaster*).

Como se pode ver nesses relatos, essa situação ocorre principalmente quando os especialistas estão no cliente, e assim a sua distância da equipe se torna um dificultador na comunicação entre ambos ou, mesmo quando esses especialistas trabalham na organização, eles podem estar envolvidos em outras atividades e, assim, não estarão suficientemente disponíveis para a equipe.

4.2.4.2. Estrutura Básica de Apoio à Produção

A efetividade da estrutura básica de apoio à produção aparece nos depoimentos dos entrevistados como uma condição que influencia a produção rápida de valor pela equipe. Uma estrutura básica de apoio, que inclui um suporte

técnico efetivo e uma infraestrutura suficiente, é uma das condições básicas para propiciar que o trabalho possa ser executado. A necessidade dessa estrutura básica de apoio pode ser vista nos depoimentos a seguir:

“Uma coisa que facilita é o próprio apoio da direção da empresa que me dá carta branca para (...) ter todo o material necessário para trabalhar. (...) não adianta nada você virar para o time e dizer ‘eu quero que vocês produzam X’ e dar uma máquina XT [computador ultrapassado] para o cara trabalhar” (Organização 3/Equipe/*ScrumMaster*).

“(...) tem várias questões burocráticas da secretaria, que às vezes não atende especificamente às nossas demandas, às vezes a gente pede uma coisa e eles negam. Umas burocraciazinhas que se for botar no papel não vale à pena, a hora de um cara trabalhando vale muito mais do que as pendengas burocráticas” (Organização 1/Equipe 1/Membro 2).

Juntamente com outras condições básicas como um ambiente de trabalho adequado e salários em dia, a estrutura básica de apoio à produção também aparece nas entrevistas como uma condição que influencia a motivação da equipe para o trabalho (veja seção 4.2.2.4).

4.2.4.3. Planejamento e sua Execução na Produção

A análise das entrevistas mostra que o planejamento do ciclo de produção e a execução desse planejamento são condições que influenciam a produção rápida de valor pela equipe. Assim, quanto mais preciso for o planejamento da *sprint* e quanto melhor for a execução desse planejamento, melhor será sua produção de valor. Um dos fatores que ajuda na precisão do planejamento é a equipe conhecer bem sua capacidade produtiva. A equipe trabalhar junta em um projeto há um certo tempo fornece boas condições para que ela conheça melhor sua capacidade produtiva, como se vê no relato abaixo:

“Eu acho que tem o próprio conhecimento. Como é uma estimativa, uma estimativa do grupo completo que já conhece o sistema, a gente tem mais ou menos uma ideia do que a gente vai conseguir ou não entregar, com o conhecimento do próprio sistema e do histórico do que a gente vinha entregando. Tipos de problema, a gente consegue comparar um com outro e fornecer uma estimativa mais certa” (Organização 7/Equipe/Membro).

Dificuldades, no entanto, aparecem quando há mudanças na composição da equipe, pois isso também altera a sua capacidade produtiva de forma ainda não conhecida pela equipe, como relata o entrevistado a seguir:

“[Ao saírem dois membros da equipe,] o que aconteceu é que a gente reduziu o escopo, mas a gente não tinha reduzido tanto o escopo, então até ter um tempo de adaptação houve um pouco de sobrecarga em quem ficou. (...) O fato de entrar pessoas também dificulta, porque são pessoas novas que não estão inseridas no projeto. Por mais especialistas e inteligentes que sejam essas pessoas, ninguém vai pegar o projeto, vai olhar o código e dizer ‘ah, isso é muito fácil, e pronto’... não existe isso, entendeu? Demora um pouco, alguns meses” (Organização 1/Equipe 1/Membro 1).

Na busca de promover um planejamento mais preciso, alguns entrevistados relatam que suas equipes criaram na *sprint* ou no *release* uma etapa adicional às indicadas pelo Scrum, onde deve ser realizada uma análise técnica mais detalhada dos itens a serem desenvolvidos, de forma a possibilitar um aumento na qualidade das estimativas do tempo de execução das tarefas necessárias para implementar os itens:

“A gente tem uma semana de *setup* [preparação] bem grande para a gente conseguir projetar, para a gente conseguir projetar testes unitários, conseguir fazer toda uma preparação para que depois consiga desenvolver de uma forma correta. (...) Antes ou durante o tempo de *setup* da versão, (...) a gente tem uma análise, um *workshop* de requisitos, não é? Nesse *workshop* de requisitos, a gente consegue antecipar, de repente, requisitos que estão implícitos, ou até que vão ser necessários para que a gente consiga entregar determinado problema pedido pelo produto” (Organização 4/Equipe/Membro).

“O que a gente faz agora é que, depois que a gente escolhe as tarefas, a gente separa um dia para sentar com outros desenvolvedores, pesquisar sobre aquilo e ver as dificuldades inerentes àquilo. (...) a gente está percebendo melhor o que a gente consegue realmente fazer naquele prazo. Então a gente está automaticamente ajustando o nosso escopo com relação à nossa velocidade de desenvolvimento” (Organização 1/Equipe 1/Membro 1).

No entanto, uma melhor análise técnica não resolve todas as questões relativas ao planejamento da produção. No Scrum, mudanças que possam afetar a meta da *sprint* em andamento devem ser evitadas a todo custo, sob o risco da *sprint* ter que ser cancelada (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). Percebe-se, no entanto, que é muito comum haver solicitações de mudanças no decorrer da *sprint*, que acabam por desviar o foco da equipe da meta estabelecida, atrapalhando o planejamento e, muitas vezes, fazendo com que a

meta não seja atingida e os resultados esperados não sejam entregues. Um erro comum relatado é a utilização de horas extras para compensar essas dificuldades e entregar o que foi prometido no prazo. Esse empenho acaba sendo confundido pela equipe com um comprometimento com as metas, com a organização ou com o cliente. No entanto, o uso de horas extras e aceleração do ritmo de trabalho de forma alguma é considerado Ágil (FOWLER & HIGHSMITH, 2001). O relato abaixo mostra bem essa confusão:

“Muitas vezes, como a estimativa não é 100% correta pode haver às vezes uma dificuldade, alguma coisa assim, ou aumentou o tempo às vezes por problemas técnicos, então nós fazemos hora extra para poder cumprir, porque nós somos comprometidos. Então nós entregamos quando nós nos comprometemos a entregar aquela versão” (Organização 3/Equipe/Membro).

Verificam-se nos relatos dos entrevistados algumas razões para que haja solicitações de mudanças após a *sprint* já ter sido iniciada, de forma a comprometer ou dificultar o cumprimento de sua meta. No relato abaixo, por exemplo, percebe-se que tanto o levantamento quanto a priorização dos requisitos que deveriam ser desenvolvidos na *sprint* não estão sendo feitos de forma adequada, já que não estão sendo levados para desenvolvimento os itens que trariam maior valor para o cliente, o que faz com que haja repriorização ao longo da *sprint*, comprometendo o planejamento:

“A gente sempre tem esses problemas de ‘olha, chegou isso aqui, isso aqui é mais urgente do que aquilo que estava acontecendo’ e fica com aquele problema de ter várias coisas correndo e nada terminado. Então o que dificulta é a falta de priorização. (...) Aquele velho mito de que tudo é prioridade, a gente sempre acha que o que está chegando agora, o cliente não pode esperar, ele tem que receber imediatamente. Acho que o que o pessoal que trabalha com a parte de negócios ainda tem uma certa dificuldade de perceber o que é que é mais importante e acha que a equipe de desenvolvimento consegue levar as coisas em paralelo” (Organização 10/Equipe/*ScrumMaster*).

De forma geral, ao efetivamente utilizar os incrementos do sistema produzidos em *sprints* anteriores que já lhe foram entregues, o cliente começa a indicar alterações importantes que devem ser feitas no sistema para melhor adequá-lo a seu uso. Um problema comum acontece quando, como o cliente já está usando o sistema no seu dia-a-dia, a realização dessas alterações ganha uma urgência tal que, na visão do cliente, não pode esperar até o final do próximo ciclo

de desenvolvimento, o que é extremamente nocivo para o planejamento da *sprint* corrente. Essa situação pode ser vista no relato a seguir:

“Há situações em que o cliente muda uma coisa no meio do *sprint*. Ele quer aquela coisa, ele quer que mude naquela hora. (...) A metodologia abomina uma situação como essa. (...) [O cliente] muda o que a gente fez no *sprint* anterior, que ele passa a usar, correto? E o *sprint* anterior acaba atrapalhando o que acontece no nosso *sprint* atual. Porque ele quer que a gente já mude. (...) As mudanças que têm dependência, a gente tem que fazer uma análise mais profunda, e (...) no meio de um *sprint*, nós não temos tempo [para isso]” (Organização 9/Equipe/Membro).

A questão de requisições urgentes no meio das *sprints* se acentua quando o cliente não compreende o processo de desenvolvimento e, assim, muitas vezes sequer sabe que não deve esperar que seus pedidos sejam atendidos imediatamente, pois isso é danoso para o projeto. É papel do *ScrumMaster* ajudar a mostrar ao cliente que, ao prejudicar o planejamento dessa forma, ele estará perdendo dinheiro (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). O depoimento abaixo mostra como esse comportamento é prejudicial ao planejamento:

“A mudança constante de requisitos, a gente não conseguia planejar. A gente planejava uma *sprint*, mas ela sempre tinha alterações e as coisas caíam de paraquedas, *on demand* [sob demanda]. O projeto foi muito *on demand*. Porque isso era a necessidade do cliente, os requisitos vinham da equipe de negócios dele. A equipe de negócios não quer saber das necessidades técnicas, eles vão pedindo e a gente tem que ir fazendo” (Organização 13/Equipe/Membro).

Quanto mais informações estiverem disponíveis para o planejamento da *sprint*, mais chances de sucesso esse planejamento terá. No entanto, alguns projetos possuem muitas dependências externas que estão fora do controle da equipe, gerando gargalos na produção, o que torna o trabalho imprevisível o suficiente para comprometer o planejamento. No relato abaixo, esse problema ocorre porque o trabalho da equipe depende dos trabalhos de equipes de outras organizações, que atuam em outras partes do projeto, além de haver uma forte dependência na infraestrutura do cliente:

“Esse *software* estava sendo construído com a equipe distribuída. Então tinham pessoas na Índia, tinham pessoas nos Estados Unidos (...). E a gente estava fazendo um pedacinho de uma coisa muito maior, o sistema era gigantesco. (...) Mas todos eles precisavam conversar. Aí você começa a colocar problemas de fuso horário

(...), problemas culturais, de língua, enfim, tudo. (...) e a gente trabalhava no ambiente de desenvolvimento da empresa [do cliente, remotamente]. (...) Aí assim, aumenta mais ainda o fator estresse. Muitas vezes a máquina lá estava fora [do ar]. Enfim, vários fatores que causavam aí o atraso” (Organização 13/Equipe/Membro).

Outro complicador para o bom planejamento é a complexidade das regras de negócio dos itens a serem desenvolvidos ou a complexidade da tecnologia envolvida. Quanto maior a dificuldade da equipe em entender o que deve ser feito e como deve ser feito, maiores são os riscos do planejamento não ser adequado, o que pode fazer com que esse planejamento não seja cumprido. As duas situações abaixo ilustram esse problema:

“O cliente [do exterior] veio para cá, passou uma semana aqui conosco, passou tudo o que eles podiam para a gente, mas é gigantesco. É uma coisa realmente imensa. Muitos conceitos que a gente não sabia. A gente fez alguns *workshops*. (...) A gente achou que estava tudo tranquilo, que dominava tudo, mas quando começou a desenvolver e começou a integrar com o sistema deles, que já existia, foi que as broncas começaram a aparecer e aí a gente viu que a coisa era muito mais complexa do que a gente imaginava” (Organização 13/Equipe/Membro).

“Dificulta ainda é que às vezes tem umas surpresas tecnológicas que a gente tem que quebrar um pouquinho mais a cabeça para poder consertar, poder andar no projeto. É uma tecnologia que a gente ainda está fazendo algumas experiências, então a gente não tem uma segurança cem por cento para não acontecer esses imprevistos ao longo do projeto. (...) Isso aí acaba às vezes levando a gente a um ponto que a gente julgava fácil, mas realmente demanda que a gente trabalhe mais para fazer e isso aí vai comer um pouco das horas aí” (Organização 12/Equipe/*ScrumMaster*).

Uma maior imprevisibilidade também está presente em projetos mais antigos que possuem código legado, ou seja, funcionalidades produzidas por uma ou mais equipes ao longo do tempo. Como se pode ver no depoimento abaixo, os maiores problemas ocorrem principalmente quando o código foi mal feito, mas o simples desconhecimento de como ele foi produzido já torna bastante difícil o trabalho de estimação do tempo necessário para criação de uma nova funcionalidade ou para a alteração de uma antiga:

“Eu tenho um problemas seriíssimos de legado. (...) O cara fez uma página de qualquer jeito, um banco de dados não normalizado, uma bagunça depois para juntar as coisas. É um pesadelo, quando você abre a página dá vontade de chorar. (...) Isso com certeza atrapalha a Agilidade. O cara não consegue entregar uma alteração, às vezes simples na página, porque ele tem que tomar todos os cuidados possíveis. Que ele mexe aqui, para outra coisa” (Organização 5/Equipe/*ScrumMaster*).

Em alguns projetos, as mudanças necessárias para o sucesso do projeto parecem ocorrer de forma mais frequente do que o tempo da *sprint* comporta, de forma que é impossível de se traçar uma meta imutável para a *sprint* e, assim, se torna difícil fazer o planejamento. O desenvolvimento de sistemas com alta dependência em legislações e regulamentações que mudam frequentemente e o desenvolvimento de *software* de apoio a campanhas de mídia são alguns dos exemplos que surgiram nas entrevistas, como aparece nos relatos abaixo:

“O que nos atrapalha hoje são as modificações legais. A gente muitas vezes tem que dar prioridade a essas coisas e tem que esquecer as outras. (...) A gente estava no meio de um *sprint* e apareceu um decreto dizendo que tinha que alterar uma coisa para a semana que vem. E eu não posso esperar o *sprint* acabar, eu tenho que liberar a coisa semana que vem, e isso atrapalha bastante” (Organização 3/Equipe/*ScrumMaster*).

“A gente tem um cliente final que tem uma necessidade de atualização muito grande, tem muitas mudanças o tempo todo. É ligado à televisão, então tem campanhas que não tem a ver com *software* e ficam vinculadas às entregas (...) muitas vezes a gente começa um *sprint* sabendo que aquilo ali não está na zona de conforto, não vai ser fácil de entregar. A gente acaba tendo que trabalhar mais que o normal, acaba tendo que trabalhar fim de semana, isso não é nada raro acontecer. (...) Em algumas situações, se a gente se recusar a entregar o todo, pode ser que metade não sirva para nada. (...) E também não adianta entregar um mês depois que aí passou o evento” (Organização 14/Equipe/Membro).

Para esses casos em que é muito difícil de se fazer o planejamento em *sprints* com metas imutáveis, o *framework* Scrum pode não ser a solução mais adequada.

4.2.4.4. Concentração dos Esforços da Equipe na Produção de Valor

A análise das entrevistas mostra que uma equipe que possui outras atribuições que não as relacionadas à produção rápida de valor será menos eficiente nessa produção de valor. Essa afirmação está de acordo com o trabalho de Womack & Jones (1998) sobre a Produção Enxuta, segundo o qual uma das condições para que se consiga colocar em prática o fluxo de valor é que o foco deve ser mantido no produto do início ao final da produção. No relato abaixo, a equipe divide seu tempo entre o projeto (produção de valor) e tarefas de suporte, o que traz problemas para o andamento do projeto:

“Se você está usando a mesma equipe compartilhada entre projeto, construção e sustentação, é claro que o negócio dá problema. Não é bom. O máximo que eu consigo me livrar disso, eu faço, mas é difícil. Porque o gerente reclama, porque o cliente pede na hora, tem que mandar o relatório e tal, e tem que parar o cara do projeto para fazer isso, e isso é um grande desafio para a gente: parar para projetos” (Organização 5/Equipe/*ScrumMaster*).

Da mesma forma, quando uma equipe trabalha em mais de um projeto, há momentos em que a produção rápida de valor de um projeto poderá ficar comprometida quando a equipe estiver dedicada a outro projeto, por exemplo, que demande maior atenção naquele momento. Nos relatos abaixo, a equipe está dedicada a diversos projetos ao mesmo tempo e a suas atividades associadas, como a solução de problemas urgentes e instalações, o que possivelmente prejudica a produção de valor em todos os projetos:

“Acontece, como a equipe é única, de interferir de urgências que surgem ou problemas ou *bug* ou acontecer alguma instalação. Interferências de outros projetos. Mas não em cima de mudanças de requisito. E sim, *bugs* e problemas que surgem. (...) a gente não tinha uma equipe dedicada” (Organização 1/Equipe 2/*ScrumMaster*).

“A gente tinha mais de um projeto em paralelo. (...) Como a gente estava envolvido em outros projetos também, a gente acabava comprometendo o projeto desse cliente. (...) A gente trabalhava até de madrugada, por várias vezes para entregar algo para o cliente” (Organização 2/Equipe/*ScrumMaster*).

Como os projetos podem ser de clientes diferentes, é grande a possibilidade de sempre haver algum cliente insatisfeito por não estar percebendo produção de valor em seu projeto.

4.2.4.5. Visão do Produto pela Equipe

No Scrum, o *Product Owner* é responsável por estabelecer a visão do produto, mantê-la e comunicá-la às partes interessadas, o que inclui a equipe de desenvolvimento (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). A análise das entrevistas mostra que, quando a equipe não enxerga claramente essa visão do produto, ela se torna menos capaz de contribuir para a distinção entre o que gera valor para o cliente e, portanto, deve ser implementado, do que não gera

e que, portanto, significa desperdício. Alguns desses problemas são relatados abaixo:

“Às vezes tem uma visão de valor que a gente pode achar que é uma coisa estratégica para o cliente, mas na verdade é uma coisa temporária só para atender a uma necessidade pequena, mas o estratégico é uma outra frente que a gente nem está vendo ainda. (...) como fornecedor de serviço, a gente tenta propor para o cliente coisas que, na nossa percepção, vão trazer valor para ele (...), então essa falta de visão estratégica prejudica que a gente seja um fornecedor ainda melhor para ele” (Organização 14/Equipe/Membro).

“A gente ficou naquela loucura de tentar resolver o que era nos passado por demanda e a gente perdeu a visão do todo. Isso era uma reclamação constante, um *feedback* constante (...) do *ScrumMaster* e da gerente, que cada um estava preocupado em fazer o seu. (...) Cada um recebia as atividades, se preocupava em fazer aquilo e não pensava no todo, não pensava (...) o que o cliente estava ganhando com aquilo. (...) No produto final, no negócio do cliente, onde é que a gente está agregando? Essa funcionalidade aqui significa o que para o cliente? Para quem vai usar?” (Organização 13/Equipe/Membro).

“A gente não se preocupa muito com os futuros dos projetos, quem se preocupa muito é o *Product Owner*. (...) A gente se preocupa com a iteração nossa. Se ele quiser mudar, pode mudar à vontade. Se não der certo, ele que se entenda com o cliente. A gente não tem uma visão de longo prazo (...) [com a falta de uma visão do produto,] as pessoas tem menos apego ao projeto, menos apego com a qualidade do projeto” (Organização 1/Equipe 2/*ScrumMaster*).

Percebe-se também, a partir dos relatos acima, que a falta de visão do produto também faz com que a equipe se sinta menos responsável pelo que está fazendo e por garantir que aquilo traga valor para o cliente, evitando mudanças que fujam à visão do produto.

4.2.4.6. Diversidade de Conhecimentos e Habilidades Dentro da Equipe

As equipes descritas por Takeuchi & Nonaka (1986) no artigo que inspirou a criação do Scrum eram multidisciplinares ou multifuncionais, ou seja, possuíam diversos conhecimentos e habilidades. De forma parecida, Schwaber (2004) descreve que as equipes do Scrum devem possuir todos os conhecimentos e habilidades necessários para gerar o valor para o cliente. O mesmo acontece na Produção Enxuta (LIKER, 2003; WOMACK & JONES, 1998), em que as equipes devem ser multifuncionais para possibilitar a produção em fluxo, sem acúmulos em lotes nem fronteiras entre departamentos especializados, o que geraria

gargalos na produção. Segundo Liker (2003), as equipes multifuncionais da Toyota propiciam um aumento de qualidade e de produtividade, permitindo a resolução de problemas técnicos difíceis, e isso é reforçado, segundo Osono et al. (2008), por treinamentos frequentes. A análise das entrevistas reforça o valor das equipes multidisciplinares ou multifuncionais para a Agilidade, conforme pode ser visto em relatos como os que seguem:

“*Layout* é uma coisa que muda o tempo todo. Então a gente tendo um time multidisciplinar que, do teu lado está a pessoa que vai fazer a mudança de *design* para contemplar tua mudança de desenvolvimento. Isso facilita bastante que a gente responda rápido. Seria mais complicado se a gente estivesse em uma situação, por exemplo, em um time que só tem desenvolvedores, e aí precisava que outra empresa fosse fazer a mudança na *layout* para contemplar o que o cliente quer, isso seria mais difícil. É mais fácil quando você tem um perfil multidisciplinar no time, que tem todo mundo necessário para fazer a mudança está no time” (Organização 14/Equipe/Membro).

“Os integrantes da equipe... como tem pessoas com conhecimentos diferentes, experiências diferentes - mais tempo, menos tempo, com pessoas que gostam de coisas separadas, eu acho que isso facilita a gente se organizar. (...) Como é a gente que se gerencia, a gente sabe o que uma pessoa tem mais facilidade de fazer e a outra não. Em que cada um tem mais facilidade. E aí isso acaba facilitando na divisão das tarefas, no que vai ser feito...” (Organização 7/Equipe/Membro).

“(...) se no meu time eu tenho o conceito de pronto que diz que o manual tem que ser atualizado, então tem que ter alguém que tenha algum perfil de documentador, algum *skill* de documentador. Se eu preciso de requisitos atualizados, alguém tem que ter um segundo perfil de analista de requisitos. Só que esse conceito de pronto direciona até como a gente vai montar o time. Se meu conceito de pronto para o projeto vai ser assim, então vou montar para atender a esse conceito de pronto. E isso faz que desde o início as entregas saiam com qualidade. (...) É o conceito de pronto que dá a coisa” (Organização 11/Equipe/*ScrumMaster*).

Nesse último relato, é importante notar como a definição de “pronto” (SCHWABER & BEEDLE, 2002; SCHWABER, 2004) orienta toda a formação da equipe multidisciplinar, de forma que todos os conhecimentos e habilidades necessários para gerar incrementos prontos do produto existam dentro da equipe.

As entrevistas mostram que um erro muito comum em equipes Ágeis é a divisão da produção entre diferentes departamentos, como *design* e testes, por exemplo, criando dependências externas para o trabalho da equipe, o que gera gargalos no processo de produção, como mostra o relato abaixo:

“A gente tem um gargalo que é o teste hoje em dia. O teste não está dentro da célula de desenvolvimento, então está meio que departamental. Tem que esperar o

dia para entregar porque o pessoal está testando outra aplicação” (Organização 4/Equipe/*ScrumMaster*).

Ao contrário dessa prática, a abordagem mais Ágil e condizente com os princípios da Produção Enxuta é a formação de equipes multidisciplinares, de forma que, nesse exemplo, os testadores deveriam se tornar membros da equipe. Para esse caso específico, a Produção Enxuta traz o conceito de *jidoka*, segundo o qual os próprios funcionários devem realizar o controle de qualidade, de forma que a qualidade deve estar embutida no processo de produção (WOMACK ET AL., 1992; WOMACK & JONES, 1998).

4.2.4.7. Uso de Conhecimentos Técnicos e Boas Práticas

Relatos dos entrevistados indicam que a equipe deve utilizar-se de conhecimentos técnicos adequados para que seja capaz de produzir o valor esperado pelo cliente. Nessa produção, devem ser aplicadas boas práticas de construção de *software* e quaisquer outras boas práticas que facilitem ou tornem possível o trabalho e melhorem a qualidade de seus resultados. No entanto, a aplicação dos conhecimentos técnicos e das boas práticas de desenvolvimento deve ter como objetivo único melhorar a produção de valor para o cliente. O relato abaixo mostra um projeto em que a equipe se preocupou demais em aperfeiçoar seus processos, mas se esqueceu da satisfação das necessidades do cliente:

“A gente estava diminuindo o percentual de retrabalho, mas o cliente não estava satisfeito. Qual era o problema? A aplicação estava com problema de performance, vários problemas. E a gente estava atacando muito no processo. Então o processo estava legal, consolidado, o pessoal estava tranquilo e tal, mas faltava talvez uma priorização do que seria mais importante para entregar de valor agora. (...). Então aquele retorno do investimento, isso aí não é feito” (Organização 4/Equipe/*ScrumMaster*).

A ênfase na excelência técnica aparece no princípio Ágil “a atenção contínua à excelência técnica e a um bom projeto aumenta a agilidade” (FOWLER & HIGHSMITH, 2001). Os depoimentos abaixo mostram essa preocupação com a aplicação de boas práticas à produção de *software*, que facilitam a sua manutenção e melhoram sua qualidade:

“O grupo teve nesse projeto uma pessoa com o perfil de arquiteto que sempre direcionava como as coisas deveriam ser projetadas para que fossem mais facilmente refactoradas. (...) Então essa proposta de *refactoring* constante faz com que eles tenham que automatizar teste, tenham que desenvolver um código bem estruturado para que facilite a manutenção dele. (...) A manutenção é já de cara, então eles vão voltar naquele código. Então isso força eles a fazer um código mais simples possível de manter” (Organização 11/Equipe/*ScrumMaster*).

“Aqui a gente tem uma preocupação muito grande com a qualidade do que a gente está gerando. (...) a gente não entrega um *software* com um código malfeito. Isso é um requisito do projeto. Porque o projeto vai ser continuado pela equipe de desenvolvimento interna da empresa para que a gente está trabalhando. (...) Se a gente fosse fazer uma coisa correndo, seria um tiro no pé” (Organização 8/Equipe/Membro).

Para buscar garantias de qualidade no trabalho realizado, diversas equipes dos entrevistados utilizam a autorregulação, ou seja, membros da equipe verificam de perto a qualidade do trabalho de seus colegas. Essa autorregulação, de acordo com Hackman (1987), é possível em equipes em que há normas comportamentais intensas e cristalizadas que a estimulem. Abaixo, dois relatos que mostram a regulação do trabalho de membros por outros membros da equipe:

“Sempre que alguém termina uma tarefa, uma outra pessoa tem que pegar aquilo que a pessoa fez e verificar se está correto. Isso envolve padrão de código, se a pessoa fez aquele negócio direito, como é que está a estrutura disso daqui. E a gente sempre tenta fazer isso da melhor maneira possível” (Organização 1/Equipe 1/Membro 1).

“Difícilmente alguém constrói alguma coisa sozinho (...). Existe sempre uma comunicação e uma validação constante de no mínimo duas pessoas. Em alguns momentos específicos a gente não envolve o time todo, mas no mínimo duas pessoas viram aquilo ali como é que nasceu” (Organização 8/Equipe/Membro).

Porém, em outros casos, o mau planejamento leva a uma aceleração anormal do ritmo de produção que, ao consumir tempo que deveria ser dedicado à preocupação com a qualidade, acaba por gerar partes do *software* produzidas com qualidade inferior, como aparece no relato abaixo:

“Como a gente trabalha muitos *sprints* com o tempo totalmente comprometido, a gente não tem margem para tratar o débito técnico. Às vezes a gente precisava ter um tempo a mais para estudar um determinado problema que resolveria melhor aquilo que a gente está querendo fazer, a gente não tem tempo hábil para pesquisar uma solução técnica melhor. Então a gente acaba acumulando débito técnico para *sprints* futuros e isso é uma coisa que atrapalha o andamento do *sprint*” (Organização 14/Equipe/Membro).

De acordo com as entrevistas, a qualidade também pode ser prejudicada em projetos que possuem código legado que, por sua vez, tenha sido produzido com baixa atenção às boas práticas e técnicas. Essa herança torna necessário refazer essas partes do *software* para garantir a qualidade do sistema como um todo.

Uma prática Ágil que aparece frequentemente nas entrevistas é a da programação em par, onde dois programadores trabalham juntos em um mesmo computador, de forma que ambos contribuam para a geração do código e para sua qualidade, facilitando ainda mais a autorregulação. Outro uso comum da programação em par é similar ao sistema OJT (*on-the-job training*, ou treinamento em serviço) da Toyota (OSONO ET AL., 2008). Nesse caso, o indivíduo é treinado pelo colega de par em algum assunto que ele desconhece, enquanto executa seu trabalho. Exemplos de relatos de usos de programação em par nas equipes dos entrevistados podem ser vistos a seguir:

“(...) a gente percebeu que programação em par *full*, para todos os itens, ajuda muito na qualidade do código, diminui o retrabalho de coisas erradas que foram para o cliente, até mesmo para homologar com o P. O., e TDD [prática Ágil de desenvolvimento com testes] complementando isso faz com que a taxa de retorno de itens defeituosos ou que foram implementados errados caiam a zero” (Organização 2/Equipe/*ScrumMaster*).

“Uma coisa que ajudou a gente... a gente tinha muita dificuldade de passar conhecimento... a gente tinha muita ilha de conhecimento, mesmo usando Scrum. (...) A gente está fazendo muita programação em par agora. Programação em par não com o objetivo de um buscar o erro do outro, mas com o objetivo de passar o conhecimento. (...) Então quando a pessoa não sabe nada sobre esse tema então ‘programa o dia inteiro com esse cara aí’. Aí ele programa e no dia seguinte já sabe” (Organização 1/Equipe 2/*ScrumMaster*).

Para que a equipe possua os conhecimentos técnicos e de boas práticas necessários para a produção de valor, muitas organizações e equipes promovem treinamentos mais formais de seus membros, de acordo com as entrevistas. Esses treinamentos são comparáveis às categorias de treinamentos da Toyota relatadas por Osono et al. (2008), que incluem o treinamento baseado em qualificação, o treinamento que tem por meta o aperfeiçoamento, e o treinamento que se baseia em conhecimentos ou habilidades. Muitas organizações contratam treinamentos externos, o que em diversos casos é limitado pelo orçamento disponível. Mas a análise das entrevistas mostra que é bastante comum o caso em que os próprios

membros da equipe ou da organização promovem treinamentos para seus colegas em assuntos que conhecem bem, como se vê nos relatos a seguir:

“A gente tem alguns seminários técnicos toda quinta feira. A gente fala de projetos e de coisas que a gente costuma fazer. (...) É o time que faz isso, toda quinta de manhã. Normalmente é alguma novidade tecnológica, algum modo novo de fazer testes” (Organização 1/Equipe 2/*ScrumMaster*).

“Essa difusão de conhecimentos geralmente a gente faz através de *dojo's*, treinamento entre a equipe ou as equipes. Quando tem uma nova coisa que vamos precisar alguém estuda e faz uma apresentação, faz para todo mundo. (...) Geralmente isso é uma demanda de acordo com a necessidade, ou alguém viu alguma coisa nova e acha legal trazer aquilo para a empresa” (Organização 7/Equipe/Membro).

“Aqui a gente tem um processo muito legal que é o treinamento semanal. (...) Eu dei um treinamento de XP [Extreme Programming, método Ágil de desenvolvimento], por exemplo. Isso mudou a cabeça de várias pessoas sobre a forma de trabalho mesmo e a forma de pensar. (...) Houve um treinamento de TDD [prática Ágil de desenvolvimento com testes] muito massivo aqui durante esse ano, que é uma prática que a gente utiliza aqui no projeto e a gente está externalizando. Durante o desenvolvimento de artefatos de outros projetos também tem usado TDD, então a qualidade como um todo acredito que está subindo” (Organização 8/Equipe/Membro).

Outra prática comum utilizada por equipes entrevistadas é a da arquitetura evolutiva, em que a arquitetura do sistema, ao invés de ser projetada no início, evolui à medida em que for necessária, ou seja, somente é projetado o que for suficiente para cada momento. Nos exemplos abaixo, um dos entrevistados relata que essa abordagem ajudou sua equipe na evolução do projeto, enquanto que outro entrevistado mostra dificuldades com essa forma de trabalhar, preferindo outras abordagens consideradas menos Ágeis:

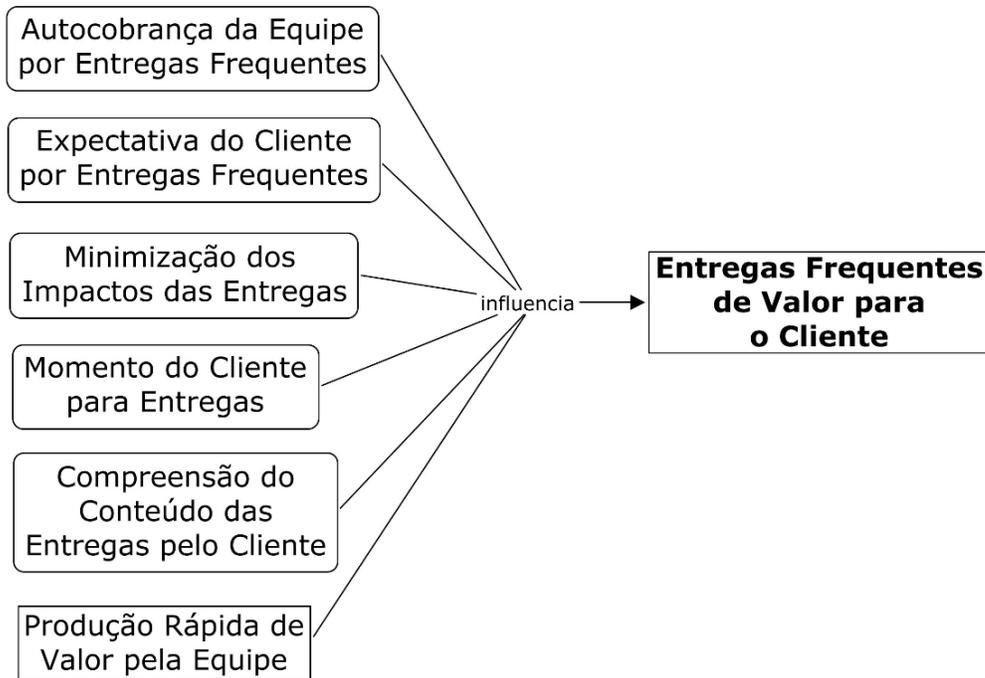
“A gente começou a fazer um design arquitetural mais evolutivo. Agora a gente tem um tempo dentro de alguns itens dentro da *sprint* para pensar melhor no *design* do projeto para poder evoluir ele a partir da implementação daquele item” (Organização 2/Equipe/*ScrumMaster*).

“Se você for pegar o processo unificado e algumas outras metodologias, a parte da arquitetura é feita no *sprint* zero, você define uma arquitetura antes. Uma coisa que dificultou as nossas entregas foi um trabalho na arquitetura durante *sprints*. Então, se a gente podia estar entregando, por exemplo, dez artefatos, a gente estava entregando seis. Porque parte da equipe estava trabalhando em requisitos não funcionais ainda” (Organização 8/Equipe/Membro).

4.2.4.8. Contribuição do Cliente na Produção de Valor

A contribuição do cliente na produção de valor, um dos fatores críticos para a prática de valores Ágeis, foi identificada a partir da análise das entrevistas como uma condição fundamental para a produção rápida de valor pela equipe. Como pode ser visto no item 4.2.7, que aborda essa questão com maiores detalhes, o cliente é considerado peça-chave nos valores e princípios Ágeis, bem como na Produção Enxuta, na determinação do que lhe traz maior valor.

4.2.5. Entregas Frequentes de Valor para o Cliente



Quadro 8: Fator crítico “entregas frequentes de valor para o cliente” e condições que influenciam sua manifestação

Para cumprir seu propósito, uma vez produzido o valor para o cliente, ele deve ser entregue. Assim, como o valor deve ser produzido rápida e frequentemente, as entregas frequentes de valor para o cliente são críticas para a prática de valores Ágeis.

Os princípios Ágeis “nossa maior prioridade é satisfazer o cliente através da entrega contínua e desde cedo de *software* com valor” e “entregar frequentemente *software* em funcionamento, desde a cada duas semanas até a cada dois meses, com uma preferência por prazos mais curtos” (FOWLER & HIGHSMITH, 2001) traduzem bem essa necessidade.

A análise das entrevistas indica que as condições a seguir influenciam as entregas frequentes de valor para o cliente. O quadro 8 sintetiza essa relação.

- Autocobrança da Equipe por Entregas Frequentes
- Expectativa do Cliente por Entregas Frequentes

- Minimização dos Impactos das Entregas
- Momento do Cliente para Entregas
- Compreensão do Conteúdo das Entregas pelo Cliente
- Produção Rápida de Valor pela Equipe

A última condição é também um dos fatores críticos para a prática de valores Ágeis identificados neste trabalho (veja a seção 4.2.5.6).

4.2.5.1. Autocobrança da Equipe por Entregas Frequentes

A partir das entrevistas, percebe-se que equipes mais engajadas nas práticas dos valores Ágeis naturalmente se cobram por fazer as entregas frequentes. Essa cobrança ocorre nos dois casos relatados abaixo, sendo que no primeiro a equipe é movida pelo orgulho do dever cumprido, e a segunda equipe é movida pela obrigação:

“(...) é a questão do próprio orgulho da equipe, sabe? A equipe se sente bem quando entrega antes, quando a gente faz um gol no primeiro tempo ainda, enfim, é bem legal. Então existe o fator motivacional de entregar antes uma coisa com qualidade” (Organização 5/Equipe/*ScrumMaster*).

“Esse mês nós temos que entregar isso e nós temos todo controle. Não temos pressão por parte do cliente. Mas temos a nossa pressão interna que é natural. Nós temos que entregar, nós somos contratados para fazer e nós temos que fazer” (Organização 9/Equipe/Membro).

Essa autocobrança ocorre principalmente porque essas equipes têm a consciência dos benefícios obtidos em cada entrega, como a geração de *feedback* para o que foi produzido e de retorno ao investimento do cliente (HIGHSMITH, 2004; LARMAN, 2003).

4.2.5.2. Expectativa do Cliente por Entregas Frequentes

Quando o cliente conhece ou está acostumado à forma de trabalho iterativa e incremental da equipe que gera entregas frequentes, é comum o próprio cliente criar expectativas quanto a essas entregas e, dessa forma, estimular a equipe a

realizá-las. Assim como na Produção Enxuta (WOMACK & JONES, 1998), a produção Ágil é “puxada” pelo cliente, ou seja, a equipe somente gera funcionalidades que foram solicitadas pelo cliente, no momento em que gerarão maior valor para este cliente. Dessa forma, é razoável crer que, na maioria dos casos, o cliente já estará esperando por aquela entrega. Nos relatos abaixo, percebe-se como essa expectativa do cliente faz com que a equipe trabalhe em função das entregas:

“Os projetos são com clientes que já tem uma mentalidade mais Ágil. Normalmente é mais fácil. Eles já querem ter um *feedback* rápido para eles poderem avaliar, fazer uma homologação além do nosso QA, olhar para o problema mais cedo e poder ter uma noção melhor dos pacotes” (Organização 14/Equipe/*ScrumMaster*).

“Então qualquer coisa que a gente mostre para ele [o cliente], telinha, protótipo, ele até fala ‘que bacana’, ele olha lá os processos mapeados, mapa mental, acha beleza, ‘nossa, que bacana, mas cadê a tela?’ Então para ele, mesmo que eu não queira, eu sou medido por isso” (Organização 5/Equipe/*ScrumMaster*).

Em alguns casos, em função do negócio do cliente, o projeto só será útil se houver entregas frequentes, e assim é natural que essa expectativa se transforme em exigência, como se pode ver no caso abaixo:

“No nosso caso aqui nesse projeto que eu trabalho, a gente trabalha com ambientes de mídia. (...) Então a entrega rápida e frequente é a única forma da gente sobreviver” (Organização 14/Equipe/Membro).

Em função disso, não é incomum encontrar projetos em que as entregas frequentes estejam previstas em contrato.

4.2.5.3. Minimização dos Impactos das Entregas

Por diferentes razões, tanto a equipe quanto o próprio cliente podem ficar apreensivos com as entregas, o que pode dificultar sua realização frequente.

Os relatos dos entrevistados mostram casos em que a equipe teme a resposta negativa do cliente caso haja algum problema com o trabalho realizado ou com a entrega em si. O medo de problemas com a entrega pode ser realmente justificado quando o ambiente em que o *software* é instalado em cada entrega é complexo e

difícil de ser reproduzido, o que diminui o controle da equipe sobre o sucesso da instalação, o que é demonstrado pelos depoimentos a seguir:

“O que dificulta um pouco é a questão da coragem. (...) Mesmo a gente tendo um ambiente de homologação que é replicado, tem ainda existe esse receio de colocar a coisa em produção e parar, de colocar em produção no meio do expediente [do cliente]” (Organização 2/Equipe/*ScrumMaster*).

“(...) muitas vezes o time tem medo de colocar em produção. (...) de dar problema, de dar um *feedback* ruim para o cliente. E dele se sentir mal por ter feito um trabalho ruim. Então você tem todo um trabalho de pegar o ambiente do cliente, produzir internamente, simular várias vezes, fazer *scripts* [roteiros executáveis] que facilitem a subida, controle de versão, ensinar... tirar os medos. (...) Então é botar o pessoal para participar para estimular isso. Isso facilita porque faz isso passar a ser uma coisa normal do projeto” (Organização 14/Equipe/*ScrumMaster*).

O cliente, por sua vez, teme os mesmos impactos negativos de entregas mal feitas. Mas, nesse caso, os problemas podem afetar diretamente o seu negócio. Clientes que, como no relato abaixo, já tiveram experiências negativas com entregas no passado, tendem a ser mais problemáticos com essa questão:

"Nem todo cliente quer receber atualizações tão constantes assim. Porque a gente veio de um passado problemático, com uma quantidade de *bugs* grande. E a gente tem aplicações que não podem parar, que se a aplicação ficar parada dez minutos, o cliente vai ter um prejuízo muito grande. Então ele tem um certo trauma de atualizações” (Organização 10/Equipe/*ScrumMaster*).

Esses problemas devem ser combatidos com soluções técnicas que facilitem as entregas, minimizando seu impacto, além da produção de *software* com qualidade, idealizado a partir de uma compreensão correta das necessidades do cliente. Uma relação de confiança deve ser criada entre cliente e fornecedor, de forma que ambas as partes superem seus medos, pois as entregas frequentes são críticas para a prática dos valores Ágeis e para o sucesso dos projetos que as utilizam.

4.2.5.4. Momento do Cliente para Entregas

Para o Scrum, as definições das datas de entregas ou *releases* ficam a cargo do *Product Owner*, que consulta o cliente para tal (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). Essas definições, no entanto, devem ser balizadas

pelos princípios Ágeis, que defendem que as entregas sejam realizadas frequentemente. De acordo com as entrevistas, porém, o cliente pode não estar disponível para entregas tão frequentes quanto a equipe desejaria ou quanto seriam recomendáveis pelos valores e princípios Ágeis. Essa dificuldade pode ser vista nos relatos a seguir:

“O que talvez tenha dificultado para o cliente é que o time em determinado momento acelerou mais do que a capacidade dele de implantar. E aí que a gente começou a usar aquela questão de planejamento de *releases*. Ele foi priorizando entregas, mas só num determinado *sprint* que ele ‘ticava’ ela para a gente liberar um pacote maior” (Organização 11/Equipe/*ScrumMaster*).

“A gente ia de mês em mês, porque no [cliente] você não pode ficar instalando versão lá facilmente, eles tem a rotina deles. (...) Burocracia do cliente mesmo, sabe? É um sistema que é homologado por toda a TI e todas as pessoas são obrigadas a usar. (...) Nesse caso, a gente costuma cumprir contrato, o contrato estipula as entregas” (Organização 1/Equipe 2/*ScrumMaster*).

“Ele vai ter todo um custo, a aplicação vai ser usada por muita gente, ele precisa de alguma forma treinar todo o pessoal dele e nem sempre ele quer receber alguma coisa de duas em duas semanas. Então às vezes a gente acaba tendo que represar algumas coisas, a gente espera o conteúdo de três *sprints* para ele poder ter uma atualização. Isso a gente vem tentando minimizar, melhorando o conteúdo de material de treinamento, de material *online*, para que, quando ele receba alguma atualização, ele saiba exatamente o que é que ele está tendo, como ele pode treinar o pessoal dele, para que possa realmente entregar o mais rápido possível alguma coisa. Ainda assim, ele vai ter o tempo dele, não pode forçar isso aí” (Organização 10/Equipe/*ScrumMaster*).

Como se pode observar, os clientes têm as suas próprias rotinas, burocracias, necessidades de treinamento, capacidades de implantação e capacidades de absorção de novas funcionalidades, às quais os momentos para realização das entregas devem ser ajustados. A grande dificuldade é fazê-lo em uma frequência suficiente para não comprometer a prática do *feedback* para a inspeção e adaptação do produto, e assim garantir a prática dos valores Ágeis.

4.2.5.5.

Compreensão do Conteúdo das Entregas pelo Cliente

O princípio Ágil “*software* em funcionamento é a medida primária de progresso” determina que as partes interessadas do projeto, entre elas o cliente, sentem e medem o andamento do projeto a partir da produção e entregas de *software* em funcionamento (FOWLER & HIGHSMITH, 2001). No entanto,

quando o cliente não compreende o conteúdo do que está sendo entregue, a sensação de avanço no projeto para o cliente pode ficar prejudicada. Dependendo do projeto, essa dificuldade de compreensão pode acontecer porque muito do que é produzido não é visível, como se pode ver no relato abaixo:

“Durante o *sprint*, você pode gerar uma série de coisas que não têm uma representação visual. A gente pode fazer todo um *sprint* de *software* que funcione que não tenha uma tela de entrada. (...) Mas esse tipo de coisa não dá noção de progresso para o cliente. (...) Se não tiver tela para fazer, ele não tem aquela sensação, não é? É um dificultador” (Organização 8/Equipe/Membro).

Uma forma utilizada por alguns entrevistados para minimizar esses problemas é, no momento da entrega, relatar com clareza para o cliente tudo o que foi gerado naquela entrega, como é feito no caso descrito abaixo:

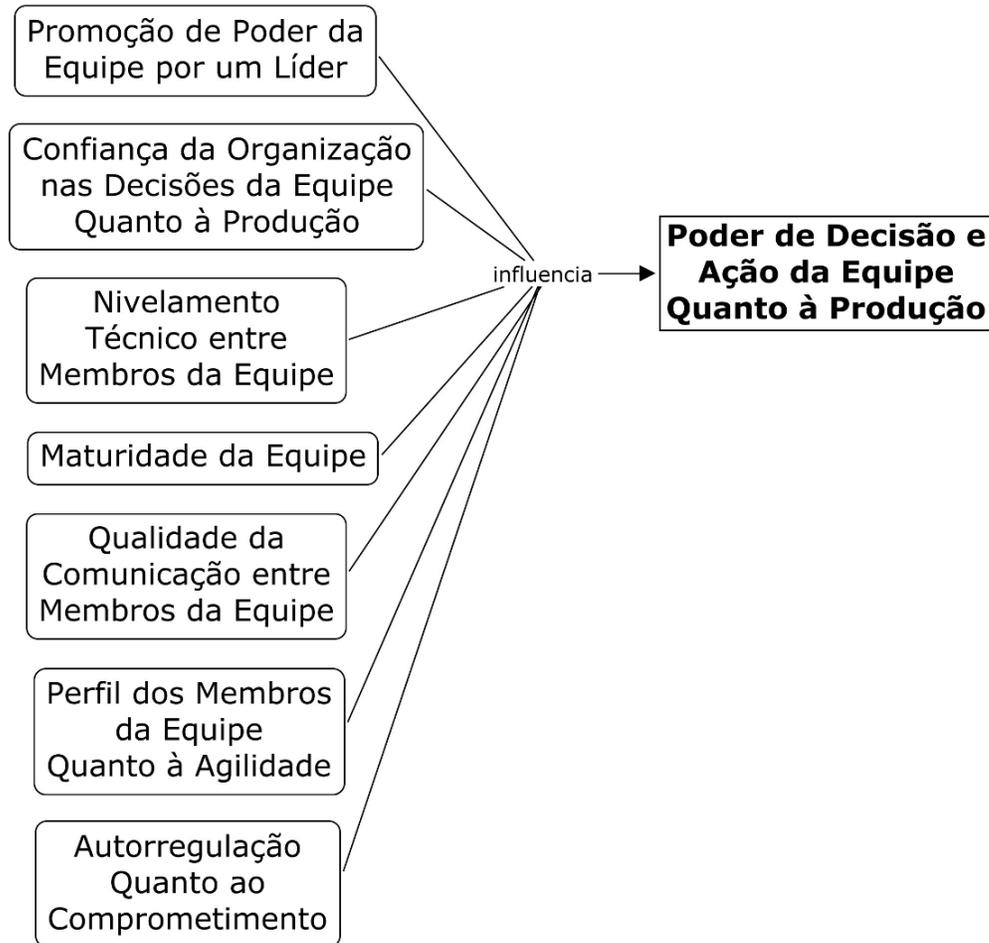
“Quando tem uma *release*, [o cliente] é informado pela área de Produto. Eles informam quais funcionalidades entraram, quais não conformidades que foram corrigidas para cada versão, isso é bem controlado. (...) É feita uma publicação. Toda vez que é publicado o nosso *software*, é informado para o cliente que foram publicadas tais funcionalidades” (Organização 4/Equipe/Membro).

Contudo, deve-se ter cuidado para que essa prática não gere uma burocracia que seja suficiente para atrapalhar o trabalho da equipe ou para desestimular a realização das entregas frequentes.

4.2.5.6. Produção Rápida de Valor pela Equipe

A produção rápida de valor pela equipe, um dos fatores críticos para a prática de valores Ágeis, foi identificada, a partir da análise das entrevistas, como uma condição fundamental para as entregas frequentes de valor para o cliente. Esse é um resultado esperado, uma vez que o valor deve ser antes produzido para então ser entregue, e com uma velocidade suficiente para permitir que essas entregas sejam frequentes. Os impactos desse fator na prática dos valores Ágeis são explicados com maiores detalhes no item 4.2.4, que aborda essa questão.

4.2.6. Poder de Decisão e Ação da Equipe Quanto à Produção



Quadro 9: Fator crítico “poder de decisão e ação da equipe quanto à produção” e condições que influenciam sua manifestação

O princípio Ágil “as melhores arquiteturas, requisitos e projetos emergem de equipes que se auto-organizam” (FOWLER & HIGHSMITH, 2001) indica que o uso de equipes autogerenciadas – ou seja, equipes que possuem o poder de ação e decisão quando à produção – é importante para a prática dos valores Ágeis, o que se confirma a partir da análise das entrevistas. Ao mesmo tempo em que reconhecem as dificuldades de sua prática, os entrevistados concordam que as equipes autogerenciadas, com autonomia de decidir sobre como irá realizar a produção e com poder de agir de acordo com suas decisões (KOZLOWSKI &

BELL, 2003; HACKMAN, 1987; MOSCOVICI, 1996), são críticas para a prática de valores Ágeis.

A análise das entrevistas indica que as condições a seguir influenciam a manifestação do poder de decisão e ação da equipe quanto à produção. Essa relação pode ser vista no quadro 9.

- Promoção de Poder da Equipe por um Líder
- Confiança da Organização nas Decisões da Equipe Quanto à Produção
- Nivelamento Técnico entre Membros da Equipe
- Maturidade da Equipe
- Qualidade da Comunicação entre Membros da Equipe
- Perfil dos Membros da Equipe Quanto à Agilidade
- Autorregulação Quanto ao Comprometimento

4.2.6.1.

Promoção de Poder da Equipe por um Líder

Autores como Cummings (1978), Kozlowski & Bell (2003) e Manz & Sims, (1987) afirmam que a função primária do líder de equipes autogerenciadas é a de facilitar esse autogerenciamento. Esse papel, em equipes que utilizam Scrum, é exercido pelo *ScrumMaster* (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). Nos depoimentos abaixo, percebem-se os *ScrumMasters* encorajando suas equipes a tomarem as decisões internamente, evitando interferir:

“Eles [membros da equipe] participam de igual para igual no cliente dessas reuniões de planejamento, então eu como *ScrumMaster* evito ao máximo de dar ordens para eles de como fazer, bem naquela ideia mesmo de liderança com colaboração” (Organização 2/Equipe/*ScrumMaster*).

“A gente procura (...) deixar a maior parte das coisas que podem ser resolvidas por eles na mão deles. Ao ponto de acabar mesmo aquela coisa de comando-e-controle. (...) Um do time fala assim: ‘olha, amanhã eu preciso levar a minha esposa ao médico’. Eu falo assim: ‘amigo, avisa ao time’. Eu deixo isso muito na mão deles” (Organização 3/Equipe/*ScrumMaster*).

A análise das entrevistas, no entanto, indica que o papel de estimular a autogerência não é simples de ser executado. Alguns fatores tornam essa tarefa

mais difícil. Quando o *ScrumMaster* possui fortes conhecimentos técnicos, por exemplo, ele se sente tentado a tomar as decisões de como o produto será desenvolvido, ou ao menos a exercer forte influência sobre isso. No relato abaixo, por exemplo, o próprio *ScrumMaster* confessa sua limitação e o consequente dano à autogerência da equipe:

“Idealmente seria o *ScrumMaster* nem ter voto, mas infelizmente eu sou um *ScrumMaster* muito técnico e eu acabo decidindo muita coisa. Eu acabo não sendo o facilitador tanto quanto eu deveria. Ao invés de eu ajudar a equipe a ser autogerenciável, eu acabo não conseguindo e falando: ‘não, vamos fazer assim’” (Organização 1/Equipe 1/*ScrumMaster*).

Nessa mesma equipe, um de seus membros indica que, na rotina de pressão de tempo para entregas, esse *ScrumMaster* também interfere na decisão de quem irá realizar o trabalho, baseado em sua percepção de quem seriam os indivíduos com os conhecimentos mais adequados para cada tipo de tarefa. Esse tipo de intervenção, como se pode ver no relato a seguir, acaba por destruir qualquer iniciativa dos membros da equipe quanto à escolha de quem realizará cada tarefa, o que compromete a autogerência:

“O [membro da equipe] não pôde pegar a tarefa que ele queria, porque existe uma pressão de tempo. E o *ScrumMaster* decidiu que seria melhor que ele não pegasse aquilo. Eu, se fosse o *ScrumMaster*, incentivaria ele a pegar, mas ele considera certas coisas mais importantes que outras. Isso acontecia de uma forma tão frequente que o [membro da equipe] e a [outro membro] simplesmente não tinham nem iniciativa de pegar alguma coisa diferente” (Organização 1/Equipe 1/Membro 1).

Em outra equipe da mesma organização, a dificuldade do *ScrumMaster* em atuar como facilitador do autogerenciamento deriva do fato de ele trabalhar também como membro da equipe. Assim, o relato mostra que, quando ele toma decisões técnicas, a equipe tende a aceitá-las sem haver debate suficiente, pois ele acaba sendo visto como o chefe, e não como membro da equipe:

“Dificulta muito mais pelo fato de eu ser o *ScrumMaster* e eu estar dentro da equipe, porque as pessoas me enxergam como o chefe. Então, para tomar decisão técnica, é difícil a equipe aceitar que ela é que deve tomar a decisão. (...) Não só as decisões técnicas. Tudo o que eu falo eles realmente aceitam. É claro que eles discutem, coisa e tal, não é assim, mandou e aceitou. Mas a minha opinião tem muito valor quando na verdade eu não queria que fosse assim. Eu queria que eles me vissem como um membro da equipe” (Organização 1/Equipe 2/*ScrumMaster*).

Um papel importante desse líder é o de remover os impedimentos que atrapalham o trabalho da equipe (SCHWABER & BEEDLE, 2002; SCHWABER, 2004). Ao estabelecer relações com outros indivíduos ou unidades da organização que podem afetar o desempenho da equipe, o líder deve ser capaz de impedir interferências externas de outras partes da organização nas decisões da equipe (DRUSKAT & WHEELER, 2003), como ocorre no caso abaixo:

“No nível que a gente chegou já agora, tem pouca interferência externa, a interferência externa é toda barrada pelo *ScrumMaster* e isso facilitou bastante” (Organização 7/Equipe/Membro).

Outro papel do *ScrumMaster*, de acordo com Schwaber & Beedle (2002) e Schwaber (2004), é o de garantir a adesão da equipe e do *Product Owner* aos valores, práticas e regras do Scrum. Nos depoimentos abaixo, pode-se ver o *ScrumMaster* exercendo esse papel ao impedir a prática de horas extras por parte da equipe, ao ensinar à equipe as práticas do Scrum e ao ensinar ao *Product Owner* como priorizar os requisitos que trazem maior valor para o cliente:

“(…) para mim, hora extra é sintoma de que foi mal planejado, sintoma de que você não está fazendo um gerenciamento correto. Tem alguma coisa errada se o time está fazendo hora extra. Então, não se faz. Ponto final” (Organização 11/Equipe/*ScrumMaster*).

“A gente consegue através do que o *ScrumMaster* nos passou, dos documentos. Nós não conhecíamos metodologias Ágeis antes e o que ele vem nos passando de um ano para cá a gente conseguiu aprender bastante. (...) Ele nos ajuda bastante, nos ensina bastante as práticas do Scrum” (Organização 4/Equipe/Membro).

“Tivemos um problema com o P. O., porque ele realmente estava muito inseguro do que ele estava fazendo. Ou seja, ele estava priorizando coisas que ele não tinha muita certeza do que era para ser feito. Então eu, como *ScrumMaster*, fiz um trabalho com o P. O. (...) para explicar para ele como priorizar os requisitos e principalmente ensinei para ele como escrever histórias e critérios de aceitação para ele ter mais certeza do que ele estava pedindo” (Organização 2/Equipe/*ScrumMaster*).

Algumas vezes, no entanto, o *ScrumMaster* deixa de lado o papel de garantir que o processo do Scrum esteja sendo seguido, o que pode desestabilizar o trabalho da equipe. Dois exemplos disso podem ser vistos nos depoimentos abaixo:

“O *ScrumMaster* teve que desenvolver. (...) Então o *ScrumMaster* começou a puxar muita responsabilidade para ele, centralizar tudo, todos os requisitos estavam na cabeça dele e também o desenvolvimento.(...) E ele ficou sobrecarregado. Então o papel dele de *ScrumMaster* começou a ser penalizado, começou a ser prejudicado com isso, ele garantir o processo” (Organização 13/Equipe/Membro).

“Eu estava muito atolado particularmente com coisa de mestrado etc. e tal, e não dava tempo de pensar no processo da equipe como um todo (...) eu não estava conseguindo ter uma visão externa da coisa. E o *ScrumMaster* é o responsável por ter manter o processo funcionando. E o *ScrumMaster* não estava conseguindo fazer mudanças, entendeu? (...) Eu acredito na acomodação do time, acho que tem sempre que ter alguém cutucando” (Organização 1/Equipe 2/*ScrumMaster*).

Nesses casos, o *ScrumMaster* está envolvido com outras atividades, internas ou externas, o que acaba prejudicando a sua atribuição principal.

4.2.6.2. Confiança da Organização nas Decisões da Equipe Quanto à Produção

A análise das entrevistas indica que o apoio e a confiança da organização nas decisões da equipe são condições essenciais para que a equipe pratique a autogerência quanto à produção. Esse resultado da análise está de acordo com a afirmação de Cohen (1993), que coloca como um dos fatores determinantes na efetividade de equipes autogerenciadas um contexto organizacional que apoie o envolvimento do funcionário através do poder de agir e tomar decisões sobre o trabalho e o desempenho dos negócios. Relatos em que essa confiança ocorre podem ser vistos a seguir:

“Eu diria que ela [a confiança na equipe de desenvolvimento] é plena. Ela é plena porque existem as especificações, o time analisa, o time vai atrás do cliente para pegar mais detalhes, o time estima, o time verifica como é que a coisa tem que colocar, o time aloca as prioridades do cliente numa *sprint*. o time decide, o time diz que não vai conseguir atender, então nesse ponto, por conta da autogestão... mas a autogestão já tem dez anos aqui. (...) A empresa começou com a autogestão há dez anos atrás. De três anos para cá eles utilizaram Scrum que já funciona muito bem com a autogestão e eu diria que isso é fator de sucesso no geral para a utilização de Scrum” (Organização 9/Equipe/Membro).

“[O apoio da alta diretoria] é fundamental. Tem divergência, mas eles dizem ‘está bom, vamos deixar os caras fazerem e vamos ver o que acontece’. O [diretor] é o defensor-mor entre eles. É o que propagou para o conselho. E o [outro diretor] é mais antigão, e aí brigava, brigava, brigava, brigava mas agora está começando a ver faturamento. E aí, ‘está bom’” (Organização 14/Equipe/*ScrumMaster*).

De acordo com as entrevistas, no entanto, é muito comum não existir essa confiança na equipe de desenvolvimento, o que atrapalha a sua autogerência e tem consequências negativas para o desenvolvimento. No primeiro exemplo abaixo, pode-se ver o discurso de um *ScrumMaster* que não confia na sua própria equipe, e atribui as dificuldades da autogerência à natureza humana:

“A gente tem correria... as pessoas acham que isso é correria. Porque as pessoas estão acostumadas com moleza. (...) A correria é muito mais psicológica. Aquela coisa assim, as pessoas sentem que trabalham doze horas. (...) Tem uma pressão dos *Product Owners* para ficar pronto, só que na verdade essa pressão não se reflete em hora extra. Mas as pessoas sentem que... às vezes as pessoas acham que trabalharam demais. (...) Então é aquela coisa: todo mundo gosta de ler *e-mail*, ver jornal, isso é um fato. Então a pessoa fica sem tempo para fazer isso e sente que trabalhou mais. Tem a pressão psicológica que aí tem a sensação que trabalhou mais” (Organização 1/Equipe 2/*ScrumMaster*).

Nesse outro caso, interferências da alta gerência limitam o poder de decisão e de ação da equipe, gerando desmotivação e comprometendo o autogerenciamento:

“A gente tinha um regime... tínhamos flexibilidade de trabalho, de hora de trabalho e um monte de coisas... que de uma hora para a outra a alta gestão cortou pelo pé e disse que todo mundo agora ia trabalhar feito fábrica. Que você tem que entrar tal hora, sair tal hora, não pode fazer hora extra, não pode isso, não pode aquilo. Então botou uma série de empecilhos que é um dos motivos de desmotivação geral da equipe. E não é só da minha célula não. (...) Se você fizer uma pesquisa interna aqui, acho que das 70 [pessoas], 70 querem sair. (...) Às vezes o problema disso é só externo. O gerente, como é um gerente de fábrica, enxerga às vezes mais fábrica do que... acha que produtividade é o cara estar sentado na mesa fazendo, às vezes isso aí complica nesse quesito. A visão de quem está de fora é que autogerenciável é descontrole” (Organização 6/Equipe/*ScrumMaster*).

Em casos extremos, ocorrem interferências diretas no dia-a-dia do trabalho da equipe. Nos relatos abaixo, verificam-se pressões da alta gerência com relação às entregas da equipe e interferências do *Product Owner* nas decisões da equipe sobre como realizar seu trabalho:

“Precisa de uma coisa para ontem, a gente faz uma estimativa e no dia da estimativa vê que não vai caber naquele prazo que é desejado e às vezes há pressões, por exemplo: ‘deixa de fazer a parte de testes’. A gente tem que entregar, ‘faz como der para entregar’ e a gente quebra o processo lá de fazer com testes (...) [As pressões vêm do] do chefe, não é? Ele que manda lá de cima. (...) Da diretoria. Sempre que tem uma pressão maior que é para ontem, vem da diretoria” (Organização 7/Equipe/Membro).

“O fato do *Product Owner*, embora esteja muito ausente, durante o *planning* [planejamento], ele tenta se meter demais nas decisões da equipe (...), qualquer tipo de decisão em relação a como a gente acha melhor fazer alguma coisa no Scrum, às vezes ele se sente mais parte da equipe do que deveria ser, de certa forma. Ele tenta influenciar demais as nossas decisões” (Organização 1/Equipe 1/Membro 2).

Em qualquer caso em que interferências externas atrapalham a autogerência da equipe, é papel do *ScrumMaster* trabalhar para que isso não ocorra (DRUSKAT & WHEELER, 2003).

4.2.6.3. Nivelamento Técnico entre Membros da Equipe

A partir das entrevistas, percebe-se que diferenças de nível técnico entre membros da equipe de desenvolvimento podem influenciar fortemente a forma como as decisões na equipe são tomadas. Um dos entrevistados chega a afirmar que, quando essa diferença é muito grande, ela impossibilita a autogerência, como se vê a seguir:

“As pessoas não conseguem ser autogerenciáveis quando se tem uma diferença nas competências muito grande entre as pessoas” (Organização 1/Equipe 1/*ScrumMaster*).

Em alguns casos, membros mais experientes não confiam em membros com menor experiência e conhecimento técnico para tomarem decisões importantes, e acabam puxando-as para si, desbalanceando o processo decisório da equipe. Essa falta de confiança pode ser vista nos seguintes depoimentos:

“A questão da confiança intertime. Alguns membros não confiam muito nos próprios membros. (...) O mais experiente não confia delegar no menos experiente. E termina querendo centralizar. Tem essa situação e tem uma outra situação. O mais experiente não quer passar para o menos experiente com medo de perder a sua posição. (...) Eu combato isso todos os dias” (Organização 3/Equipe/*ScrumMaster*).

“A disparidade de conhecimento técnico dificulta muito e às vezes se você pedir a opinião de alguém da equipe, não vai conseguir te dar, ele não tem capacidade de te dar *feedback*” (Organização 1/Equipe 1/Membro 2).

Assim, de acordo com Langfred (2007), essa baixa confiança entre os membros da equipe pode reduzir a autonomia individual e desacoplar ou diminuir

a interdependência entre tarefas na equipe, fazendo assim com que a equipe se organize de forma ineficiente.

Um resquício de formas tradicionais, não Ágeis, de se trabalhar é a existência de líderes técnicos nas equipes, como descrita nos relatos abaixo:

“Uma coisa que atrapalha é aquela figura do líder técnico, do camarada mais antigo querer ditar determinadas regras para o resto do time. E o time termina aceitando porque o cara é mais antigo, está no projeto desde o começo e ninguém vai muito contra a opinião dele, então quando ele coloca a opinião primeiro, o resto vai atrás. (...) Se ele falou primeiro, ninguém vai contra e todo mundo fala amém” (Organização 3/Equipe/*ScrumMaster*).

“Mas eu acredito que a equipe ser auto-organizável é muito difícil. Porque na prática, (...) mesmo que a pessoa não seja chefe, sempre tem um líder técnico, que as pessoas seguem. (...) As pessoas não tomam decisões sozinhas. (...) É auto-organizável para o chefe, mas dentro da equipe existe uma hierarquia, uma hierarquia de conhecimento que leva a uma hierarquia de decisões” (Organização 1/Equipe 2/*ScrumMaster*).

Nota-se que esses líderes técnicos tendem, por iniciativa própria ou por iniciativa da equipe, a ditar como o trabalho deve ser feito e a tomar as decisões importantes, prejudicando a autogerência da equipe.

4.2.6.4. Maturidade da Equipe

A maturidade individual dos membros da equipe e a maturidade coletiva, enquanto equipe autogerenciada, aparece nas entrevistas como uma condição que influencia o poder de decisão e ação da equipe. Nos relatos abaixo, a falta de maturidade se mostra como um obstáculo à tomada de decisões e, portanto, à autogerência:

“A gente teve uma deficiência que nem todos os integrantes da equipe tinham uma maturidade de desenvolvimento que pudesse tomar algumas decisões de projeto. (...) É claro que liberdade vem com responsabilidade, não é? E é por isso que eu falo também da deficiência de maturidade do profissional como é que pode impactar, não é? Você dá a mão e ele quer o braço, e aí é extremamente complicado também, não é?” (Organização 8/Equipe/Membro).

“Isso daí [maturidade da equipe] é uma coisa que a gente não está muito bem não, o nosso time ainda não está muito auto-organizável não. (...) Em alguns momentos, o time deixa determinadas coisas rolaem porque aí fica um esperando pelo outro e ninguém faz nada. Então a gente, como *ScrumMaster*, tem que dar uma espetada” (Organização 3/Equipe/*ScrumMaster*).

De acordo com a opinião e experiência de alguns dos entrevistados, a maturidade como equipe autogerenciada, no entanto, vai se desenvolvendo com o tempo, conforme a equipe vai trabalhando junta em ambientes que propiciem condições para tal, como relatado abaixo:

“As pessoas têm que entender que, se elas são livres no quesito de ter uma autonomia para tocar o projeto, para se envolver com as coisas, elas têm que ser responsáveis também. No começo eles ainda achavam que, se desse alguma coisa errada, era culpa do *ScrumMaster*, ou do P. O., ou dos sócios da empresa. Com o tempo, eles foram percebendo que não, que a responsabilidade era deles mesmo, então eles tinham que desenvolver um trabalho melhor” (Organização 2/Equipe/*ScrumMaster*).

“(…) essa interação dessa equipe toda, todo dia, isso faz com que a pessoa... ela se automelhora por ela mesma. Na reunião diária, todo mundo diz o que fez e o que vai fazer. E isso fica muito evidente quando tiver um membro da equipe que não está nesse passo. Então geralmente ele mesmo... fica tão evidente, que ele vai pegando rumo” (Organização 3/Equipe/Membro).

“O que dificulta é mais maturidade da própria equipe. Algumas pessoas do projeto não trabalhavam num ambiente assim, então não é fácil fazer uma mudança desse tipo de uma hora para a outra. (...) Eles não tinham um ambiente em que tinha confiança em que eles poderiam ser auto-organizáveis, autogerenciáveis. Então um pouquinho de choque. Em alguns momentos eu tenho que falar alguma coisa para a pessoa fazer. Em alguns momentos a bola cai na questão da pró-atividade. Então esse é um ponto que dificulta, mas acho que com o tempo isso se resolve, isso é mais vivendo em projetos onde as equipes são assim, isso aí tende a melhorar” (Organização 12/Equipe/*ScrumMaster*).

Nesse último caso, por exemplo, fica clara que experiências profissionais anteriores dos membros da equipe podem causar impactos na adoção da autogerência. Assim, aqueles que estão acostumados a trabalhar em ambientes em que controlam ou são controlados, como é a prática mais comum, apresentam as maiores dificuldades.

4.2.6.5.

Qualidade da Comunicação entre Membros da Equipe

A qualidade da comunicação entre membros da equipe é, de acordo com a análise das entrevistas, uma condição que influencia o poder de decisão e ação da equipe. A existência de um canal aberto de comunicação entre os membros da equipe influencia positivamente no seu poder de decidir como equipe, pois

aumenta a visibilidade do trabalho sendo realizado, como se pode ver no relato abaixo:

“O que facilita é um bom canal de comunicação que a gente tem, que a gente sempre deixa bem explícito o que está acontecendo e o que vai acontecer (...), todo mundo sabe qual o próximo passo e aí todo mundo discute” (Organização 1/Equipe 1/Membro 2).

Porém, a capacidade de se comunicar a e orientação à comunicação depende também de características dos indivíduos envolvidos. Em seu depoimento, um dos *ScrumMasters* afirma que acredita ser própria dos profissionais de tecnologia da informação uma dificuldade com a comunicação, como se pode ver a seguir:

“O que costumo dizer para os nossos *ScrumMasters* é que o time tende a não se comunicar. É uma tendência natural da nossa área. Você pode colocar um na frente do outro no computador que ele vai abrir um *chat* [canal de mensagens instantâneas] para falar com o cara. Então ele tende a usar os piores meios de comunicação, mesmo quando o cara está do seu lado. (...) Me parece que é uma coisa muito normal do perfil da nossa área, de ser mais reservado” (Organização 11/Equipe/*ScrumMaster*).

Independentemente desse fator, a qualidade do relacionamento entre membros da equipe, além de um forte motivador, emerge das entrevistas como importante aspecto para a qualidade dessa comunicação, e, portanto, para a autogerência da equipe. O bom relacionamento entre membros da equipe, de forma a melhorar a capacidade de se trabalhar junto eficientemente, faz parte dos modelos mentais estabelecidos por Druskat & Pescosolido (2002) como importantes para a efetividade da autogerência da equipe, o que pode ser visto nos relatos abaixo:

“Enquanto equipe, nós cinco da célula, é uma maravilha. Não tem problema nenhum. A gente interage bem, conversa muito. Um confia muito no outro para que o trabalho seja desenvolvido.” (Organização 6/Equipe/*ScrumMaster*).

“Um sabe que pode contar com o outro e um sabe onde o outro é melhor. Então automaticamente eles se organizam de maneira que cada um pode contribuir mais para o todo. (...) É bem quase estilo parceiro de truco, sabe? Aquela piscadinha de olho, aquela coçadinha, aqueles sinaizinhos, eles já conhecem. E se um está com um problema, o outro vem, sabe e ajuda” (Organização 5/Equipe/*ScrumMaster*).

Já segundo Moscovici (1996), a interação entre membros da equipe é o que mais influi em como as atividades são conduzidas e em seus resultados. Nos

exemplos acima, de acordo com Moscovici (1996), as interações ocorrem no nível socioeconômico. De acordo com esse autor, essas interações influem positiva ou negativamente nas interações que ocorrem no nível de tarefa. Nos depoimentos abaixo, essa influência é positiva:

“O que facilita? A integração da equipe. (...) A equipe é muito integrada. Tanto os desenvolvedores, quanto os *testers* [testadores] e os analistas de negócio, a gente trabalha em sintonia, então quando um termina o outro já começa, e vai fazendo o que pode trabalhar em paralelo etc. etc. Isso é muito legal” (Organização 9/Equipe/Membro).

“A gente procura conversar antes de desenvolver. (...) A gente procura tomar o mínimo possível de decisões sozinho. Por quê? Não é falta de confiança, é uma validação espontânea” (Organização 8/Equipe/Membro).

Indivíduos entrevistados (como por exemplo, Organização 3/Equipe/Membro) destacam com alguma frequência a importância das reuniões do Scrum (planejamento, diária, revisão e retrospectiva) como facilitadores da comunicação face a face. Essa forma de comunicação é considerada por Daft et al. (1987) como o meio mais rico de comunicação, e considerada por Straus & McGrath (1994) e Straus (1997) como responsável por aumentos de produtividade. Um dos entrevistados destaca a importância para a autogestão dos membros participarem de todas as reuniões estabelecidas, e os efeitos negativos de quando isso não acontece:

“(…) há *sprints* em que você não tem como participar de uma reunião por uma questão lógica. Como essa minha, por exemplo, que eu estou no cliente agora. Então eu não tenho como participar de reuniões. Então, se eu tivesse alguma coisa, algum ponto de vista, eu acabo não expondo porque não estou lá. Aí depois, quando eu sei do que aconteceu, aí ‘não seria melhor assim, por exemplo?’. ‘Ah, seria, mas você não estava aqui’. E aí é aquela coisa, não é? Acho que a parte ruim da autogestão, quando as reuniões são tão fundamentais que não dá para perder nenhuma. E todo mundo sabe que não dá para comparecer a todas, porque existem diversas coisas, não é?” (Organização 9/Equipe/Membro).

A análise das entrevistas fornece indicações de que a melhor forma de se promover a comunicação face a face, defendida por Daft et al. (1987), Straus & McGrath (1994) e Straus (1997), é a equipe de desenvolvimento trabalhar no mesmo local, compartilhando junta o máximo das suas horas de trabalho. Em seguida, alguns relatos que destacam a importância da equipe trabalhar junta para possibilitar a autogerência:

“Como a gente está tudo no mesmo local físico, a gente consegue ter as reuniões diárias, as reuniões de abertura, todas as reuniões... até a análise, quando tem alguma análise bem detalhada para fazer com o produto... conversar face a face, tu não precisa fazer eletronicamente, a gente não tem nenhuma dificuldade quanto a isso” (Organização 4/Equipe/Membro).

“Acho que a co-locação é essencial para isso. Quando tem que separar geograficamente a equipe... não que seja impossível fazer Scrum com isso, mas acho que dificulta muito, já que é uma coisa que é tão focada em comunicação. A gente tem a equipe todo mundo junto, também o P. O. junto, fica todo mundo na mesma sala” (Organização 10/Equipe/ScrumMaster).

“O que facilitou foi o time estar junto (...), o time está ali numa redoma que é só dele, um ambiente que é só dele, eles estão sempre interagindo, os horários dos componentes do time são os mesmos para que eles estejam interagindo o tempo todo” (Organização 11/Equipe/ScrumMaster).

No entanto, nem sempre é possível a equipe trabalhar junta, na mesma hora e no mesmo local. Existem os casos em que as equipes trabalham distribuídas, ou seja, cada membro ou grupo de membros trabalhando em locais distintos. Esses casos, no entanto, não foram abordados neste trabalho. Porém, mesmo quando trabalham no mesmo local, algumas equipes relatam problemas com horários de trabalho diferentes entre seus membros, o que diminui sua comunicação e prejudica a autogerência. Regimes diferentes de trabalho entre os membros, como estágios, trabalhos de horário parcial e integral etc. aparecem como os principais motivos para que esses problemas ocorram, como se pode ver nos depoimentos a seguir:

“(...) a gente não consegue ficar também todo dia oito horas o pessoal junto aqui para poder discutir o que a gente está fazendo, conversar. (...) Volta e meia a gente tem alguns dias onde um não trabalha de manhã, trabalha à tarde, às vezes a gente tem inversão de horário mesmo. (...) Tem gente que é realmente meio período, não fica realmente oito horas. (...) A gente (...) vai tentar resolver isso no próximo ano. Mas aí são mais questões administrativas e financeiras da empresa” (Organização 12/Equipe/ScrumMaster).

“O que dificulta mais a comunicação é o seguinte: os funcionários *part-time*. Tempo parcial. Então normalmente eles não trabalham quatro horas todo dia. Eles fazem oito horas hoje, oito horas amanhã, quatro horas...” (Organização 1/Equipe 2/ScrumMaster).

“Alguns membros tem horários diversos. (...) O tempo que a gente tem todo mundo junto não é o mesmo. Na verdade, eu disponho de seis horas durante o dia para fazer essas coisas com todo mundo junto, e isso dificulta. (...) Tem um membro que tem filho, e aí o filho precisa entrar na escola e ela não pode chegar antes de nove horas porque não tem ninguém com quem deixar e por causa disso ela entra uma hora mais tarde e sai uma hora mais tarde. Tem outro que mora numa outra cidade

que não tem condução para ele chegar aqui antes” (Organização 3/Equipe/*ScrumMaster*).

Muitas equipes relatam que o próprio ambiente de trabalho é formatado para facilitar a comunicação e, assim, promover a autogerência e a prática de valores Ágeis. Alguns exemplos podem ser vistos nos relatos abaixo:

“(…) recentemente teve algumas mudanças no local de trabalho do time, aí todo o time está em duas bancadas, uma do lado da outra. Todo mundo, para falar com a outra pessoa, simplesmente vira para o lado e chama sem problema. Ficava um pouco espalhado e ai tinha silos do time separados. Isso facilitou mais porque, você tendo uma pessoa do teu lado do seu time para você trocar uma ideia ou, de repente, só discutir uma solução rapidamente, você não precisa levantar e atravessar a sala para ir falar com ela, de repente parar o que você está fazendo, fica uma coisa mais Ágil” (Organização 14/Equipe/Membro).

“Aqui a gente tem os quadros brancos espalhados pela empresa, então o pessoal discute bastante em frente a estes quadros, face a face. Até mesmo a disposição das mesas: os desenvolvedores ficam um de frente para o outro, então facilita bastante essa comunicação. (...) a gente já montou toda a empresa num local novo visando exatamente isso. (...) O ambiente aqui também a gente tem uma área fora onde as pessoas descansam melhor, (...) antes não tinha um lugar onde as pessoas poderiam ter um momento de descompressão. Então o próprio ambiente (...) proporcionou um jeito da equipe desestressar” (Organização 2/Equipe/*ScrumMaster*).

“[Facilita] o ambiente já ser integrado e as pessoas não terem portas entre si. Então não há divisórias” (Organização 9/Equipe/Membro).

O posicionamento adequado dos postos de trabalho, a inexistência ou diminuição de barreiras físicas entre os membros da equipe e a existência de quadros que podem ser usados livremente pela equipe são alguns elementos presentes nesse tipo de ambiente.

4.2.6.6. Perfil dos Membros da Equipe Quanto à Agilidade

A partir da análise das entrevistas, há indicações de que o perfil de membros da equipe (e sua vivência profissional, que ajuda a formar seu perfil) exerce uma importante influência sobre a efetividade da autogerência e da prática de princípios Ágeis como um todo. Entrevistados relatam terem enfrentado diversos problemas com indivíduos contratados que trouxeram vícios de mercado contrários à Agilidade:

“Tem muitas pessoas que eu vejo que não são totalmente prontas para o conceito de time autogerenciável. Não sei se é um pouco da cultura de pessoas que já estão trabalhando há alguns anos com modelo de um gerente ou coordenador que pede para eles fazerem as coisas e fica aquela estrutura de *command e control* [comando e controle]. (...) Não é tão fácil o cara se transicionar para esse modelo de time autogerenciável” (Organização 14/Equipe/Membro).

“(...) a gente percebeu que é melhor trazer gente que é recém-formada e capacitar elas do que contratar gente de fora, para poder praticar os princípios de uma forma mais efetiva, mais clara, entender melhor porque aquilo é importante. (...) O grande problema nosso foi que essas pessoas, no começo, por virem - entre aspas - contaminadas de outras empresas, de outro tipo de pensamento, (...) então a gente teve um certo problema para elas entenderem como trabalhar dessa forma mais livre, mais autônoma” (Organização 2/Equipe/*ScrumMaster*).

“Ter pessoas com um perfil adequado ao que se está se propondo a trabalhar. (...) Existem perfis que encaixam com que o Ágil precisa, e às vezes existem perfis que não encaixam. Primeiro, é a questão da pessoa que acha que a técnica é tudo, eu acho que isso não encaixa. Tem pessoas que são extremamente técnicas, extremamente ligadas a ferramentas (...). Eu tenho visto poucas pessoas que têm a visão do Manifesto Ágil” (Organização 8/Equipe/Membro).

Essas questões influenciam diretamente no processo de contratação de membros para equipes Ágeis. Em diversos casos presentes nas entrevistas, ou se trazem indivíduos sem experiência de mercado e, portanto, sem seus vícios, ou se contratam indivíduos acostumados com a forma Ágil de se trabalhar ou, no mínimo, verifica-se se os perfis dos possíveis contratados são compatíveis com a Agilidade.

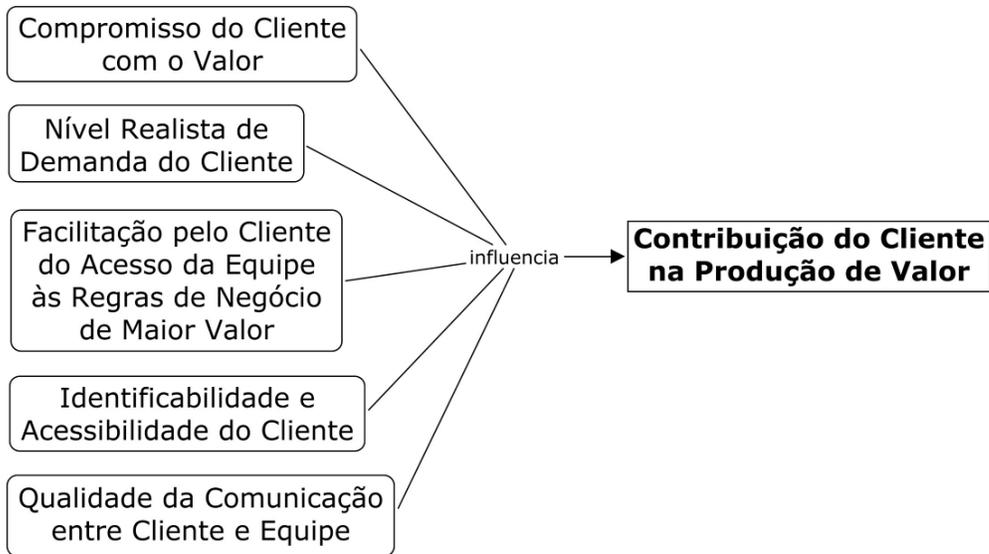
4.2.6.7. Autorregulação Quanto ao Comprometimento

De forma geral, uma condição básica para que a autogerência aconteça é a própria equipe garantir o comprometimento de seus membros com o projeto e com os processos. Uma situação em que a equipe se autorregula, por exemplo, quanto ao compromisso com as reuniões diárias, pode ser vista abaixo:

“O que facilita é uma certa pressão pras pessoas cumprirem horário e estarem aqui no *daily meeting* [reunião diária ou *daily scrum*] e não ‘fanfarronarem’, a gente ter o *daily meeting* de manhã pras pessoas chegarem na hora e a gente ter uma certa cobrança para ninguém fanfarronar. Essa cobrança vem de exemplos, liderança e uma pessoa do próprio time reclamar quando a pessoa fanfarrona, aquele negócio de expor as pessoas” (Organização 1/Equipe 1/*ScrumMaster*).

Essa autorregulação, que combate individualismos e desorganização, é possível acontecer, segundo Hackman (1987), quando o grupo possui normas comportamentais intensas e cristalizadas que a estimule.

4.2.7. Contribuição do Cliente na Produção de Valor



Quadro 10: Fator crítico “contribuição do cliente na produção de valor” e condições que influenciam sua manifestação

A contribuição do cliente na produção de valor pela equipe de desenvolvimento é, de acordo com as entrevistas, crítica para a prática dos valores Ágeis.

O valor Ágil “colaboração com o cliente mais que negociação de contratos”, aliado ao princípio Ágil “pessoas de negócio e desenvolvedores devem trabalhar em conjunto diariamente por todo o projeto” (FOWLER & HIGHSMITH, 2001), sugerem que a contribuição do cliente na produção de valor é importante para o sucesso do projeto. Assim, na produção de *software*, o nível de participação dos clientes na sua execução deve ser alto, de acordo com a classificação de Zeithaml & Bitner (2003), pois afeta diretamente os resultados produzidos.

Da mesma forma, na Produção Enxuta, o cliente é tratado como parte integral do processo de produção (WOMACK ET AL., 1992). A produção de valor deve ser “puxada” pelo cliente, de forma que o produto que ele deseja seja produzido e entregue quando ele necessita (WOMACK & JONES, 1998).

A análise das entrevistas indica que as condições abaixo influenciam a contribuição do cliente na produção de valor pela equipe. O quadro 10 mostra essa relação.

- Compromisso do Cliente com o Valor
- Nível Realista de Demanda do Cliente
- Facilitação pelo Cliente do Acesso da Equipe às Regras de Negócio de Maior Valor
- Identificabilidade e Acessibilidade do Cliente
- Qualidade da Comunicação entre Cliente e Equipe

4.2.7.1. Compromisso do Cliente com o Valor

O princípio Ágil “nossa maior prioridade é satisfazer o cliente através da entrega contínua e desde cedo de *software* com valor” estabelece que o principal objetivo da prática de valores Ágeis é a geração de valor para o cliente (FOWLER & HIGHSMITH, 2001). Assim, a noção de que o cliente deve estar comprometido com a obtenção de valor do projeto ou desenvolvimento que ele contratou parece bastante óbvia. No entanto, surgiram nos relatos dos entrevistados casos em que o cliente ou seu representante não necessariamente estão comprometidos com a geração de valor, de forma não contribuem ou pouco contribuem na produção de valor. De acordo com o relato a seguir, o interesse maior do cliente ou de seu representante, o *Product Owner*, é simplesmente “queimar” o orçamento, sem que isso lhe traga problemas ou ameaças:

“(…) por alguma razão obscura - nem vou entrar nesse mérito, de quem são os interesses - às vezes o cara não quer colocar em produção. Ele quer demorar com aquilo. (...) Tem vários motivos. Muitas vezes não tem interesse naquele projeto. Não tem interesse em ver o projeto dar certo. É mais fácil queimar orçamento e pegar um outro orçamento e já está justificado porque foi de um outro fornecedor. Muito vezes é medo de... o principal *anti-pattern* de P. O. é ele proteger a própria situação. Ele está querendo se manter ali na ilusão de que está com estabilidade e segurança e ele não quer receber nenhum tipo de *issue* [problema] porque senão ele vai perder o emprego” (Organização 14/Equipe/*ScrumMaster*).

Nesse outro depoimento, o entrevistado relata situações em que o cliente pode até ter interesse no valor, mas não se envolveu o suficiente no projeto para garantir o valor ou escolheu mal seu representante, o *Product Owner*, que pode não estar comprometido com os resultados do projeto:

“Quando o cliente está participando, está bom porque rapidinho ele diz que não está bom, rapidinho ele diz como ele quer, então rápido ele já muda tudo então tem que adaptar. Agora há aqueles clientes que vêm para a reunião - que a gente convida para a reunião - olham dizem ok, ok durante quatro meses e depois dizem que não era o que queriam. Acontece esse tipo de situação também. Mas isso é uma característica do envolvimento do cliente com o que ele quer. Quando eu digo ‘o cliente’, (...) pode ser (...) o *Product Owner* do cliente, que ele escolheu. Ele escolheu um *Product Owner* que não está comprometido com o sucesso dele. É uma coisa que nós não temos como controlar” (Organização 9/Equipe/Membro).

Esse exemplo ilustra a importância tanto do envolvimento do cliente na produção de valor quanto da escolha de um representante que esteja comprometido com a geração de valor. O *ScrumMaster*, nesse caso, deve trabalhar para aumentar o envolvimento do cliente ou deve ajudar na troca do *Product Owner* (SCHWABER & BEEDLE, 2002; SCHWABER, 2004).

4.2.7.2. Nível Realista de Demanda do Cliente

As demandas excessivas do cliente sobre a equipe de desenvolvimento, de acordo com as entrevistas, são uma contribuição negativa do cliente na produção de valor. Essas demandas se traduzem na forma de requisições urgentes durante a *sprint* de desenvolvimento, imposição de produção de mais itens aos que a equipe pode se comprometer em produzir ou demandas de suporte decorrentes da própria interação com o cliente.

Primeiramente, essas demandas podem provocar na equipe um ritmo exagerado de trabalho, de forma a criar os desperdícios *muri* (sobrecarga nas pessoas ou equipamentos além dos limites naturais) e *mura* (desbalanços no ritmo de produção) que, de acordo com Hampson (1999), podem diminuir a qualidade e gerar diversos outros problemas, inclusive de saúde em membros da equipe. No exemplo abaixo, pode-se ver a demanda excessiva do cliente gerando sobrecarga no trabalho da equipe:

“(...) a gente passou seis meses fazendo hora extra, trabalhando final de semana, trabalhando feriado. (...) Foi muito mais pela característica de onde os requisitos surgiam, da raiz lá dos requisitos, que vinham do cliente, da equipe de *marketing* e negócios do cliente. (...) todo mundo estava no mesmo barco tentando ganhar o cliente” (Organização 13/Equipe/Membro).

No entanto, de acordo com Womack & Jones (1998), quando a produção “puxada” é efetivamente executada, as demandas dos clientes tendem a se tornar muito mais estáveis, já que os clientes sabem que podem conseguir o que desejam imediatamente.

4.2.7.3.

Facilitação pelo Cliente do Acesso da Equipe às Regras de Negócio de Maior Valor

Para produzir os resultados que o cliente espera, a equipe de desenvolvimento deve ter acesso às regras de negócios dos itens que gerarão maior valor para o cliente. Verifica-se, pelas entrevistas, que esse acesso é facilitado quando há pessoas no cliente especificamente designadas para o projeto, disponíveis para a equipe quando, por exemplo, ela necessita tirar alguma dúvida ou receber o *feedback* sobre algo que foi entregue. Os relatos abaixo ilustram essa situação:

“O que facilita é ter do lado do cliente, alguém especificamente voltado para o projeto. (...) facilita muito (...) o *feedback* do cliente, não é? Não adianta a gente entregar em duas semanas e ele enrolar duas semanas para dar o *feedback* daquela versão” (Organização 8/Equipe/Membro).

“Esses *stakeholders* são funcionários da empresa [cliente], são pessoas que são envolvidas no projeto, são as “galinhas” e eles estão sempre disponíveis. (...) A gente mantém o contato por telefone o dia todo e às vezes até por videoconferência pelo *Skype* [programa de comunicação por voz-sobre-ip]. Então essas pessoas durante a *sprint* auxiliam a equipe tirando dúvidas, explicando os procedimentos, mantendo sempre esse contato alinhado com o P. O.” (Organização 2/Equipe/*ScrumMaster*).

Quando, no entanto, o cliente ou seu representante não dedicam tempo suficiente para o projeto, eles podem não estar contribuindo para a produção de valor o quanto seria necessário para a prática efetiva dos valores Ágeis. Nesses casos, a equipe sente dificuldades no entendimento do que é o valor que deve ser produzido naquele ciclo de desenvolvimento. Essas questões aparecem nos depoimentos abaixo:

“Por não estar no mesmo ambiente, muitas vezes o P. O. está ocupado, resolvendo outro problema, ou está no telefone, ou ter saído da empresa para ir num cliente deles, então em algum momento acontecem esses problemas de comunicação (...) Os dois P. O.’s que já trabalharam com a gente do cliente, eles exercem funções do negócio deles dentro da empresa deles” (Organização 2/Equipe/*ScrumMaster*).

“Tem clientes que não são muito cooperativos, não dão tempo, e nesse caso a gente tem que fazer as vezes do cliente. Então ou o supervisor ou o coordenador pesquisa, aí ele tenta extrair do cliente melhor as informações e a gente faz o resto” (Organização 5/Equipe/*ScrumMaster*).

Nesses casos, com frequência os requisitos mais prioritários do *product backlog* não chegam à reunião de planejamento com um nível de compreensão suficiente para que possam ser desenvolvidos (SCHWABER & BEEDLE, 2002; SCHWABER, 2004), ou seja, não chegam preparados pelo *Product Owner* para tal, como pode ser visto nos relatos abaixo:

“Já chegou caso de reuniões, cara, que é impressionante. (...) [O P. O.] chegava para gente aqui: ‘o que você acha que a gente tem que fazer nisso?’. (...) ‘O que vocês acham, dá uma ideia aí, gente!’. Na reunião, o *Product Owner* não sabia o que queria! Aí complica. (...) Em teoria, eu não aceito mais nada que o pessoal não sabe o que é. Em teoria, a equipe teria que falar não. Mas para o [diretor], a equipe não tem [coragem] para falar não” (Organização 1/Equipe 2/*ScrumMaster*).

“O *proxy* [representante] para o cliente é totalmente o *Product Owner*, que tem 10% de dedicação ao projeto. (...) Tem *sprint* que tem um nível de incerteza enorme, (...) [que] vem da falta de preparo dos *Product Owners* em pedir coisas, eles não seguem o *Product Backlog* direito e vem com coisa de última hora mal definida, e depois somem depois da reunião de *planning* [planejamento]” (Organização 1/Equipe 1/*ScrumMaster*).

Como em geral atua como um intermediário, o *Product Owner* traz para a equipe sua visão sobre o que ele interpreta que deve trazer maior valor para o cliente. Os entrevistados relatam casos em que o *Product Owner* pode não representar bem esse cliente (ou clientes), prejudicando assim a contribuição do cliente na produção de valor, como se pode ver nos depoimentos abaixo:

“Às vezes eu sinto que o P. O. não representa muito bem esses usuários, que às vezes o que ele acha que seria muito bom, nem sempre é bom. Existe uma coisa do *feeling* pessoal dele, mas se for uma questão de *feeling* pessoal, cada um da equipe tem *feeling* pessoal...” (Organização 1/Equipe 1/Membro 1).

“(...) você às vezes não tem o contato direto com o cliente. (...) por bem ou por mal, o P. O. ainda vai dar aquela visão dele com relação ao que vai ser feito. (...) Eu queria ver ele ainda mais presente no cliente do que ele está hoje” (Organização 10/Equipe/*ScrumMaster*).

“(...) será que isso aí [o produto] está interessando aos clientes que já estão no portal? (...) A gente não sabe como está o retorno do investimento. (...) Isso aí [o *feedback* do cliente] fica na área de produto e isso é como te falei, eles não estão com um processo muito firme nisso. Então a gente não sabe se o que a gente está

entregando tem realmente valor ao cliente em geral” (Organização 4/Equipe/*ScrumMaster*).

Para muitos clientes, é difícil a identificação do que lhe traz maior valor, o que prejudica o trabalho da equipe, como se pode ver no relato abaixo:

“O cliente não tinha experiência em como conduzir um projeto de *software*, então ele teve bastante dificuldade no começo para definir o que ele queria, o que seria uma coisa de valor” (Organização 2/Equipe/*ScrumMaster*).

Nos relatos seguintes, devido a essa dificuldade, o cliente pede funcionalidades que fogem à visão estabelecida do produto. No entanto, as equipes buscam mostrar para o cliente que essas funcionalidades não agregam valor ao produto, ajudando o cliente nesse processo de identificação de valor:

“Como nosso escopo é sempre negociável, a gente tem facilidade de conversar com o cliente e falar: ‘você tem certeza que isso é importante? Isso aí você vai precisar para agora ou vai ser uma coisa que você vai precisar para o futuro?’ (...) Então essa facilidade de comunicação com o cliente é boa nesse aspecto, a gente consegue filtrar bem o que vai ter valor ou não” (Organização 14/Equipe/Membro).

“O que vinha ganhava prioridade, o que mudava era desenvolvido, mas sempre validando se aquela ideia estava dentro da visão inicial do produto para que o produto não se desvirtuasse do objetivo dele. Então eu, como o *ScrumMaster*, e mais a atuação do P. O., tinha essa política de ‘ok, registramos a ideia, mas isso faz parte da visão inicial do produto?’. ‘Faz’, se não faz, a gente vai fazer, mas vai negociar” (Organização 11/Equipe/*ScrumMaster*).

No segundo caso, fica aparente que, se o cliente ainda assim insistir, a equipe acaba por produzir a funcionalidade solicitada, podendo gerar, dessa forma, desperdício.

No entanto, a equipe deve entender que as mudanças propostas pelo cliente, desde que dentro da visão estabelecida do produto, devem ser aceitas como algo natural e bem-vindo no processo de desenvolvimento, como está descrito no valor Ágil “responder a mudanças mais que seguir um plano” e no princípio Ágil “mudanças de requisitos são bem-vindas, mesmo tardiamente no desenvolvimento. Processos Ágeis utilizam a mudança em favor da vantagem competitiva do cliente”. Porém, como acontece em alguns casos, a aceitação da mudança foi problemática para a equipe descrita no depoimento abaixo:

“(...) alguns desenvolvedores não aceitavam o cliente mudar de opinião sobre alguma coisa que já estava feita. Então eles ficavam bravos: ‘nossa, mas o P. O. não resolve o que ele quer, a gente acabou de fazer uma coisa de novo, ele tem muita dúvida’” (Organização 2/Equipe/*ScrumMaster*).

Esse comportamento é comum em grupos acostumados com o desenvolvimento em cascata (MCCONNELL, 1996), contrário aos princípios Ágeis.

4.2.7.4. Identificabilidade e Acessibilidade do Cliente

Duas dimensões de características dos clientes destacaram-se nos relatos dos entrevistados quanto a sua identificabilidade e acessibilidade. Na primeira, diferenciam-se clientes internos e clientes externos. Na segunda, os clientes são tais que o projeto deve atender a somente um cliente, deve atender a vários clientes ou deve atender ao mercado, ou seja, o projeto é um produto que é ou será vendido no mercado.

Nos casos em que o cliente é interno, sua contribuição para a produção de valor é em geral mais facilitada, pois esse tipo de cliente é mais acessível que clientes externos, como se pode ver no relato abaixo:

“Com o cliente interno, a gente consegue trazer ele pras reuniões. Ele percebe como é que está o andamento do projeto. No *review* [revisão da *sprint*], ele participa, então ele consegue ver como é que o projeto está andando, quais as dificuldades que ele está tendo, o que ele está precisando de novo e isso facilita bastante para a gente. Realmente participa junto com a gente na hora de entregar o produto” (Organização 10/Equipe/*ScrumMaster*).

Quando, no entanto, o desenvolvimento deve atender a vários clientes, o *Product Owner* deve saber priorizar os itens de maior valor com equilíbrio entre os diversos interesses, de forma a melhor atendê-los. Os depoimentos abaixo mostram bem as dificuldades envolvidas nesse trabalho:

“Pode ser que ele [um dos clientes] esteja esperando alguma coisa e (...) pode ser que (...) aquele pedido (...) fique de fora. (...) A gente tenta minimizar isso percebendo se aquilo tem uma prioridade muito alta para o cliente. Às vezes é um pouco difícil a gente perceber isso, porque quando o cliente pede alguma coisa para você, ele sempre vai dizer que é superimportante. Já aconteceu da gente ter uma versão e que a gente inicialmente esperava que alguma coisa ia chegar para ele e acabou no fim das contas não chegando, porque coisas aconteceram. A gente tenta

avisar a esse cliente o mais rápido possível” (Organização 10/Equipe/*ScrumMaster*).

“Muitas vezes a gente tem essa dificuldade a atender às necessidades do cliente porque, como a gente tem uma gama muito grande de clientes, cada um está num contexto diferente. (...) Tenho que atender em cima dessa realidade. Às vezes a gente cria uma coisa muito grande para um que não atende a outro. Então a gente tem que ir filtrando as necessidades do cliente, de uma forma que seja o produto uma coisa o mais genérica possível para todos os contextos” (Organização 6/Equipe/*ScrumMaster*).

Outro caso comum ocorre quando o produto é vendido para clientes no mercado, não havendo assim, clientes facilmente identificáveis. Esse caso é ilustrado pelo relato a seguir:

“Um dos problemas que nós tivemos quando a gente começou a pensar em adotar o Scrum foi definir quem é o cliente. Na verdade, nós temos dezesseis mil clientes. Mas nós temos aqui dentro da empresa um setor que representa esses dezesseis mil. (...) O nosso termômetro são os clientes. (...) A gente acompanha os *bugs* que são reportados pelos clientes todo mês. Também, as ligações que os clientes nos fazem, dando o *feedback* deles. (...) quando a coisa não está boa, a quantidade de *bugs* reportados para a gente aumenta, e até o cliente ligando para o suporte dizendo que estão insatisfeitos porque aquela versão que colocou deu um monte de problema” (Organização 3/Equipe/*ScrumMaster*).

Nesses casos, é comum haver na empresa um departamento responsável por interpretar e traduzir para o *Product Owner* as necessidades e solicitações do mercado com relação ao produto.

4.2.7.5. Qualidade da Comunicação entre Cliente e Equipe

De acordo com as entrevistas, a qualidade da comunicação entre cliente e equipe é decisiva na contribuição do cliente na produção de valor.

Alguns grupos utilizam meios eletrônicos e sistemas para compensar a distância do cliente e aumentar a comunicação, como nos depoimentos a seguir:

“Mas o que a gente sempre faz tem uma comunicação diária, essa comunicação acontece ou por *Skype* [programa de comunicação por voz-sobre-ip] ou por *e-mail*. (...) a gente trabalha dessa forma, sem ficar lado a lado, mas tendo realmente diariamente uma interação entre a gente” (Organização 12/Equipe/*ScrumMaster*).

“Como o cliente fica lá, o que a gente faz é muita comunicação por telefone e *e-mail* e todos os PMs [gerentes de projeto] do cliente vêm aqui. A gente faz uma tarde ou uma manhã de reunião. (...) E fora isso é telefone, *e-mail*, *issue tracker*

[sistema de acompanhamento de erros] está sendo usado. *Burndown chart* [*sprint burndown*]. Boa *issue tracker*. Porque o que acontece? Eu não posso levar os *post-its* para lá. Então a gente não usa *post-its* no quadro, a gente usa o *Jira* [sistema de acompanhamento] como *taskboard* [quadro físico de tarefas], a gente está com o *plugin* do *Greenhopper* [*software* adicional ao *Jira* para uso com *Scrum*]. (...) O cliente já se acostumou em acompanhar” (Organização 14/Equipe/*ScrumMaster*).

“Nenhum desenvolvedor gosta de falar com cliente. Isso é um fato. Eu acho que 90% dos desenvolvedores. (...) O *MSN* [programa de mensagens instantâneas] ajuda um pouco nisso. (...) As pessoas tem muita dificuldade de ligar, ligar é um caso muito extremo. (...) Ligar é uma coisa intrusiva. Agora, no *MSN* você manda a mensagem e a pessoa responde. (...) Facilita muito. Se não fosse isso, iria dificultar bastante o acesso” (Organização 1/Equipe 2/*ScrumMaster*).

Nesse último caso, o *ScrumMaster* tem a opinião que os desenvolvedores não gostam de ter contato direto com o cliente e, assim, utilizam meios eletrônicos para compensar isso. No entanto, outros relatos identificam como nocivo o uso dos meios eletrônicos como substitutos para a conversação face a face com o cliente, como se pode ver no relato abaixo:

“Com o cliente, tem algumas situações em que a gente se acomoda um pouco em ficar falando pelo *GoogleTalk* [programa de mensagens instantâneas], comunicadores instantâneos em vez de falar com o cliente pessoalmente mesmo. Em algumas situações, você se comunicar pessoalmente, resolve muito mais rápido” (Organização 14/Equipe/Membro).

Diversos entrevistados, no entanto, consideram o contato com o cliente ou com o usuário final como prejudicial para o trabalho da equipe. Os relatos abaixo ilustram casos em que esse contato é considerado prejudicial:

“A gente tinha um problema muito grande, que a gente tinha muito contato com o cliente. Só que a coisa era muito zoneada, porque tinha gente que ligava de cinco em cinco minutos pedindo coisas, pedindo coisas, pedindo coisas e estava dificultando o desenvolvimento, as coisas não andavam porque no meio de uma coisa pedia outra, então foi criada uma barreira para tentar blindar um pouco a equipe de desenvolvimento para trabalhar em paz” (Organização 1/Equipe 2/*ScrumMaster*).

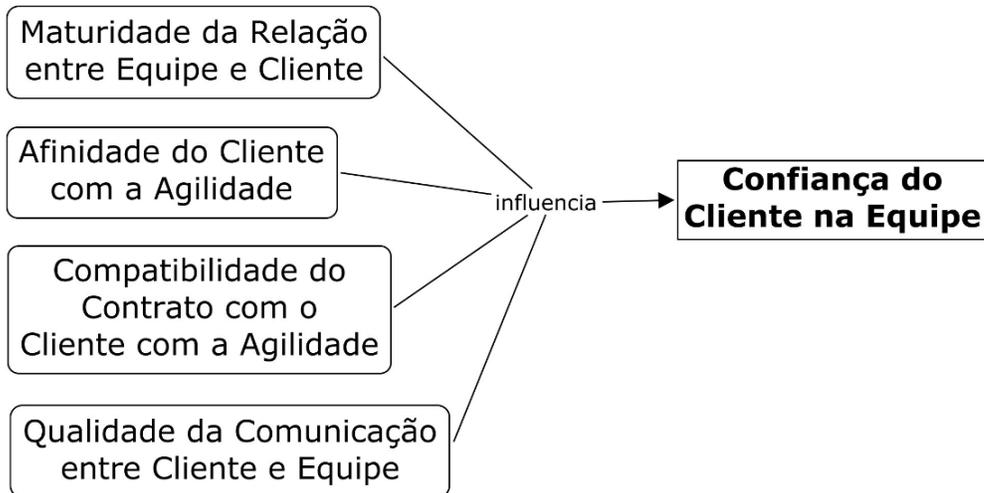
“Só o pessoal de negócios pode fazer [contato com o cliente], o pessoal de desenvolvimento não pode. A gente botou essa regra senão vira bagunça. (...) O usuário só tem, de gente, os olhos e mesmo assim de cachorro. Então, brincadeiras à parte, eles são muito carentes... (...) Às vezes acontece o seguinte: o desenvolvedor atende um chamado e resolve mais rápido que o cara de suporte. Porque o cara de suporte vai seguir todo um *proforme* para poder fazer um atendimento. O cara de desenvolvimento não, vai lá, abre a função, altera e resolve. Então muitas vezes acontece de, quando você dá uma mãozinha, o cara quer a mão, braço, a perna... então (...) a tendência é que o cliente pule o suporte e queira ligar

direto para o desenvolvimento. (...) Aconteceu aqui e foi um dos motivos que a gente precisou blindar um pouco a equipe de desenvolvimento” (Organização 6/Equipe/*ScrumMaster*).

“A gente está naquele problema do suporte de terceiro nível. Que isso aí, por ter uma comunicação tão aberta, entram coisas que talvez a gente não devesse estar fazendo. (...) Às vezes o excesso de comunicação, muita coisa perguntando, tu interrompe teu trabalho. Então tem que achar um meio-termo nisso” (Organização 4/Equipe/*ScrumMaster*).

O problema mais comum é o cliente, devido à proximidade, achar que pode fazer solicitações à equipe, sejam de suporte ou para realização de novas funcionalidades, e acaba por prejudicar todo o planejamento da *sprint* de desenvolvimento. É papel do *ScrumMaster* buscar meios de evitar que isso aconteça (SCHWABER & BEEDLE, 2002; SCHWABER, 2004).

4.2.8. Confiança do Cliente na Equipe



Quadro 11: Fator crítico “confiança do cliente na equipe” e condições que influenciam sua manifestação

O valor Ágil “colaboração com o cliente mais que negociação de contratos”, aliado aos princípios Ágeis “nossa maior prioridade é satisfazer o cliente através da entrega contínua e desde cedo de *software* com valor” e “pessoas de negócio e desenvolvedores devem trabalhar em conjunto diariamente por todo o projeto” (FOWLER & HIGHSMITH, 2001), sugerem que cliente e equipe devem ter uma relação próxima para o sucesso do projeto. Os resultados da análise da pesquisa sugerem que a confiança do cliente na equipe é um fator crítico para o estabelecimento dessa relação próxima e, portanto, para a prática de valores Ágeis.

As seguintes condições influenciam a confiança do cliente na equipe de desenvolvimento. O quadro 11 sintetiza essa relação.

- Maturidade da Relação entre Equipe e Cliente
- Afinidade do Cliente com a Agilidade
- Compatibilidade do Contrato com o Cliente com a Agilidade
- Qualidade da Comunicação entre Cliente e Equipe

4.2.8.1. Maturidade da Relação entre Equipe e Cliente

O tempo de relação entre o cliente e a equipe e as experiências que viveram juntos em trabalhos anteriores é, de acordo com a análise das entrevistas, uma importante condição que influencia o nível de confiança entre eles. Esse resultado da análise está de acordo com Zeithaml & Bitner (2003), que afirmam que um dos grandes benefícios de um relacionamento de longo prazo com um cliente é que se criam sentimentos de confiança do cliente no prestador do serviço. Essa relação de confiança estabelecida a partir de um histórico de relacionamento profissional é ilustrada nos relatos abaixo:

“É facilitada pelo fato de a gente ter um cliente um pouco diferente. (...) A gente tem uma relação diferente com o cliente. (...) A gente tem uma relação de confiança. É uma confiança meio desconfiada, mas é uma relação de confiança. Às vezes o cliente fica meio ‘bolado’ (...), mas tem um histórico de relação, então isso facilita um pouco” (Organização 1/Equipe 2/*ScrumMaster*).

“Tem um cliente de longa data, de, sei lá, sete anos. (...) A coisa agora está num ritmo Scrum (...) bem direitinho. Só que foi um trabalho longo com o cliente” (Organização 14/Equipe/*ScrumMaster*).

No entanto, quando o cliente é novo, várias barreiras contra a Agilidade podem ser levantadas, como exemplifica o depoimento abaixo:

“O cliente precisava de um pouco de segurança, no sentido de ter um escopo bem definido do que ia ser entregue e seguindo todo um processo de qualidade. [Essa necessidade de segurança] vem por ser um cliente novo, tipo era o primeiro projeto com a [organização] e ele não confiava plenamente na competência da empresa. Era um projeto onde a gente estava estrategicamente ganhando e conquistando aquele cliente. Então isso é um fator importante, a gente teve que literalmente dar o sangue aqui para conquistar esse cliente, e ele não fazia questão de disfarçar que ainda não confiava, a gente precisava ganhar a confiança deles” (Organização 13/Equipe/Membro).

Nesse caso, o cliente parece acreditar que, ao forçar mais trabalho para que a equipe conquiste sua confiança, mais valor estará sendo gerado para ele em menos tempo e, dessa forma, a equipe provará sua competência. No entanto, essa crença é falsa, pois a sobrecarga na equipe acaba por gerar desperdícios (LIKER, 2003).

4.2.8.2.

Afinidade do Cliente com a Agilidade

As entrevistas revelam uma maior facilidade de se estabelecer uma relação de confiança entre a equipe Ágil e o cliente quando ele conhece, aceita e já teve ou deseja ter experiências com a prática de Agilidade com seus fornecedores, onde ele deverá ter uma maior participação no desenvolvimento projeto. No entanto, essa aceitação depende de características individuais do cliente e assim, de acordo com Zeithaml & Bitner (2003), a empresa fornecedora deve procurar atrair clientes que se sentirão confortáveis com essa forma de trabalhar para que o projeto seja bem executado. Por exemplo, quando predomina no cliente a cultura tradicional de projetos, que se caracteriza pelo pouco envolvimento do cliente durante a execução do projeto, a equipe parece encontrar maiores dificuldades em estabelecer essa confiança, ao menos no início da relação. Um exemplo de problemas desse tipo pode ser visto no relato abaixo:

“[Temos clientes] com uma mentalidade mais de projetos fechados mesmo. Principalmente com clientes muito grandes. Banco, por exemplo. Então tem um certo problema. Assim no sentido de confiar que a gente tem uma proposta de *software* que a gente conhece e que ele não conhece naturalmente por não estar envolvido com *software*. Porque é tradicionalmente comando e controle” (Organização 14/Equipe/*ScrumMaster*).

Zeithaml & Bitner (2003) afirmam que uma forma de se melhorar a atuação dos clientes nos projetos é educá-los, preparando-os para o processo do serviço, como fez a equipe do entrevistado abaixo:

“A gente percebeu que dava para fazer mais, que o cliente tinha muitas dúvidas, que não sabia como priorizar, que ele não entendia esse negócio do que retorna mais valor, do que retorna menos valor. Então (...) a gente resolveu (...) treinar o cliente” (Organização 2/Equipe/*ScrumMaster*).

Essa abordagem, no entanto, pode ser eficiente em alguns casos e falhar em outros, principalmente quando a cultura predominantemente tradicional de projetos é muito forte no cliente.

4.2.8.3.

Compatibilidade do Contrato com o Cliente com a Agilidade

Características do contrato estabelecido entre cliente e a organização em que está inserida a equipe influenciam são influenciadas pela confiança entre equipe e cliente. De acordo com as entrevistas, os contratos que melhor refletem a forma Ágil de se trabalhar e, assim, possibilitam uma melhor prática dos valores Ágeis são os contratos de escopo aberto, nos quais escopo do projeto vai sendo definido durante a sua execução, e contratos os baseados em entregas de valor (geralmente, também de escopo aberto), em que a organização é paga por cada entrega que realiza. Exemplos dos dois tipos de contrato podem ser vistos nos relatos abaixo:

“(…) quando a gente precisa abrir mão de alguma coisa que estava no *sprint* para entrar outras coisas de mais valor, ou então renegociar o escopo do que pode ficar para depois dentro do *sprint*, o que realmente é mais importante, a gente tem um diálogo muito bom com o cliente nesse aspecto. Não é aquele tipo de contato que empedra que o escopo é aquele ali e que foi fechado e não pode mudar. Então essa facilidade de negociação e de conversa com o cliente. (...) Tudo o que a gente faz de conversas e planejamento de escopo é com o objetivo comum de entregar bem o produto que o cliente quer para que ele fique bem com o cliente dele e a gente fique bem com as nossas entregas e todo mundo fique satisfeito no processo” (Organização 14/Equipe/Membro).

“(…) nosso contrato é baseado em entregas de valor. (...) Esse cliente já estava muito com o pé atrás com empresas de desenvolvimento de *software*. (...) Quando o cliente percebeu que a gente estava colocando mais responsabilidade para o nosso lado e dando mais segurança para ele através do contrato, ele topou fazer um teste e aí logo nos primeiros meses ele já percebeu uma diferença no jeito de desenvolver o *software*, e começou a ter retorno sobre o investimento que ele estava fazendo” (Organização 2/Equipe/*ScrumMaster*).

Nesse último exemplo, o cliente decidiu experimentar o contrato baseado em entregas de valor e, ao ver seus resultados positivos, iniciou uma relação de confiança com a equipe.

Outros clientes, no entanto, exigem ou preferem contratos de escopo fechado, como ocorre no depoimento abaixo:

“[O cliente] não [quer] dividir o risco no sentido de não confiar na gente, quando é um cliente novo ou quando nunca teve esse tipo de projeto com nenhuma outra empresa. ‘Mas cara, como assim vocês querem fazer dessa forma? Por que não pode ser escopo fechado? Para mim é muito melhor!’ Aí existe um trabalho comercial de explicar por que é muito pior. Por que vai ser muito mais caro. Até que o cliente fala ‘está bom, então não vou comprar’. Se rola de outra forma, vamos ganhar dinheiro, vamos fazer *software*. Dentro de uma margem de segurança, mas é o que acontece” (Organização 14/Equipe/*ScrumMaster*).

Nesse tipo de contrato, o escopo do projeto é definido em seu início, dificultando a aceitação de mudanças ao longo do projeto. Assim, quanto maior o nível de detalhamento desse escopo, mais difícil será a prática do valor Ágil “responder a mudanças mais que seguir um plano” (FOWLER & HIGHSMITH, 2001).

4.2.8.4. Qualidade da Comunicação entre Cliente e Equipe

A qualidade da comunicação entre cliente e equipe, de acordo com a análise das entrevistas, assim como influencia a contribuição do cliente na produção de valor, também é uma condição para o estabelecimento de uma relação de confiança entre cliente e equipe. O item 4.2.7.5 aborda essa questão com maiores detalhes.