

A difícil concepção de sistemas informáticos

Renata Bastos Ferreira (ATAN) renata.bastos@atan.com.br
Francisco de Paula Antunes Lima (UFMG) fpalima@dep.ufmg.br

Resumo

A indefinição do escopo de projeto, comumente atribuída à dificuldade do cliente saber o que quer, gera retrabalho e produtos inadequados à utilização real. Em pesquisa conduzida em uma empresa de automação, explicamos os problemas em torno do escopo (1) pela impossibilidade do usuário do futuro sistema conhecer, a priori, as funcionalidades; (2) pela falta de contato do analista com os diversos agentes envolvidos na utilização das soluções de automação; (3) devido às dificuldades metodológicas para explicitar as necessidades do usuário e o saber prático dos operadores, (4) pela diferença de linguagens entre analistas e usuários do sistema e, finalmente, (5) pela inversão de papéis entre cliente e fornecedor. Propomos princípios metodológicos associando abordagens descendentes e ascendentes, que permitem obter informações mais adequadas e relações de cooperação entre fornecedor e cliente, facilitando o reajuste permanente dos requisitos de projeto.

Palavras-chave: Escopo, Especificação de Requisitos, Explicitação do Saber, Automação.

1. Introdução: definição do escopo de projeto

Um problema freqüente da concepção de sistemas informatizados é a dificuldade de se definir, de início, o escopo do projeto (funcionalidades, requisitos etc.). No momento da definição do escopo, o cliente e os diversos usuários do sistema geralmente não sabem o que querem, isto é, parecem ser incapazes de definir detalhadamente as funções e procedimentos operacionais do sistema, o que comumente é atribuído ao desconhecimento, por parte do cliente, (1) da tecnologia de informática (linguagem técnica e possibilidades das aplicações), (2) de características do próprio processo a ser informatizado ou (3) de suas próprias necessidades, sempre cambiantes ao longo do desenvolvimento do sistema. Em consequência, os programas desenvolvidos não atendem ou atendem mal às necessidades dos clientes, gerando frustração, perda de tempo em retrabalho e, inevitavelmente, conflitos e desgaste.

De modo geral, a tendência atual de diversificação de produtos para atender mercados fragmentados, mediante uma maior customização, depende do reconhecimento preciso das necessidades reais dos usuários. Essa adequação pode não ocorrer devido a problemas de comunicação, quando se adota um processo de desenvolvimento sequencial, separando a fase de concepção da operação em escala real. As consequências são bem conhecidas: «canibalismo» entre produtos semelhantes, não atendimento das necessidades dos clientes, retrabalho, atrasos e perda de produtividade em geral. Em seus estudos de caso, Wheelwright e Clark (1992) revelaram o modo reativo de resolução de problemas adotado pelos gerentes de projeto («after-the-fact» *problem solving*), ao invés de prevenção de problemas por meio de pré-projetos e planejamento mais efetivos. Esse comportamento reativo diminui o poder dos gerentes para influenciar as soluções de projeto.

O problema do escopo mal definido acontece em todos os tipos de projeto, de bens de consumo às instalações e equipamentos industriais, de natureza mais técnica. Por exemplo, a

análise dos conflitos entre clientes e fornecedores de equipamentos automáticos na França revelou o papel crucial das especificações («*cahier des charges*»), que estavam sujeitas a múltiplas interpretações. De um lado, os clientes estavam insatisfeitos com equipamentos que não atendiam suas necessidades, de outros os construtores reclamavam da imprecisão dos requisitos das encomendas. Após estudos conduzidos por um consórcio entre clientes, fornecedores, entidades de classe e pesquisadores, chegou-se à conclusão que a especificação não é uma questão puramente técnica (a ser resolvida, por exemplo, com um *check list* aperfeiçoado), uma vez que os problemas decorriam, sobretudo, das relações sociais entre cliente e fornecedor (TIGER, 1990).

No caso dos projetos de automação estudados por nós, um dos técnicos associa os prejuízos sofridos pela empresa à precariedade da especificação dos requisitos de projeto: *"O foco principal do prejuízo não está no desenvolvimento técnico do projeto, pois este é muito bom, mas no pouco contato com o cliente e na especificação rápida"* (programador). O foco de análise deve, portanto, ser redirecionado para as relações entre fornecedores e clientes.

2. Quadro de referência: associando abordagens descendentes e ascendentes

Os métodos tradicionais de engenharia possuem em comum o fato de serem um processo descendente (*top-down*), que parte de uma lista de requisitos ou do «conceito» de um produto, progressivamente concretizado ao longo do processo de desenvolvimento do produto (PDP). Isso torna o PDP, predominantemente, seqüencial e linear. Mesmo os reajustes, nos momentos de iteração e de avaliação, são correções de percurso em função de um objetivo pré-determinado, não influenciando de modo significativo o escopo inicial. A distância entre o produto oferecido e as necessidades reais dos clientes e usuários, verificada na prática, coloca em questão a eficácia desse processo seqüencial. *"Uma outra abordagem, que pode ser qualificada de ascendente ou bottom-up, parte do princípio de que a consideração das condições de realização da atividade de trabalho, desde as etapas iniciais de um projeto, pode ajudar a esclarecer as escolhas a serem feitas em relação à concepção dos sistemas técnicos e dos postos de trabalho. Trata-se de uma abordagem complementar à abordagem descendente ou top-down"* (DUARTE, 2000).

O princípio desta abordagem é o inverso da anterior: as fases subseqüentes influenciam nas decisões tomadas em fases iniciais, o que instaura uma temporalidade específica ao PDP, dando mais força às linhas de retroação. Assim, a possibilidade de (re)definir o escopo durante o PDP depende das diversas combinações entre estratégias de projeto descendentes e ascendentes, onde se criam condições de antecipação e de correção, sem necessidade de um grande retrabalho. Os esquemas abaixo permitem compreender melhor as combinações.

A figura 1 representa as duas abordagens e o espaço possível para a consideração de determinantes das situações futuras de utilização do produto (operação, manutenção...). A área sob a curvas indica o espaço de manobra para a equipe de projeto efetivar alterações e encontrar soluções alternativas, o que depende da capacidade de antecipação dos compromissos a serem estabelecidos no decorrer do PDP.

Na figura 2 estão representadas diferentes possibilidades de articulação entre as abordagens ascendentes e descendentes. A **linha A** é típica do PDP tradicional, quando as informações sobre as situações reais de utilização aparecem apenas no final do projeto ou após a partida. O espaço de manobra para se estabelecerem compromissos favoráveis é reduzido, com efeitos negativos sobre as condições de trabalho e sobre a produtividade. A **linha B** representa um PDP em que as abordagens ascendentes e descendentes, apesar de iniciarem simultaneamente, só se encontram ao final do projeto, sem interações durante o desenvolvimento. Seus

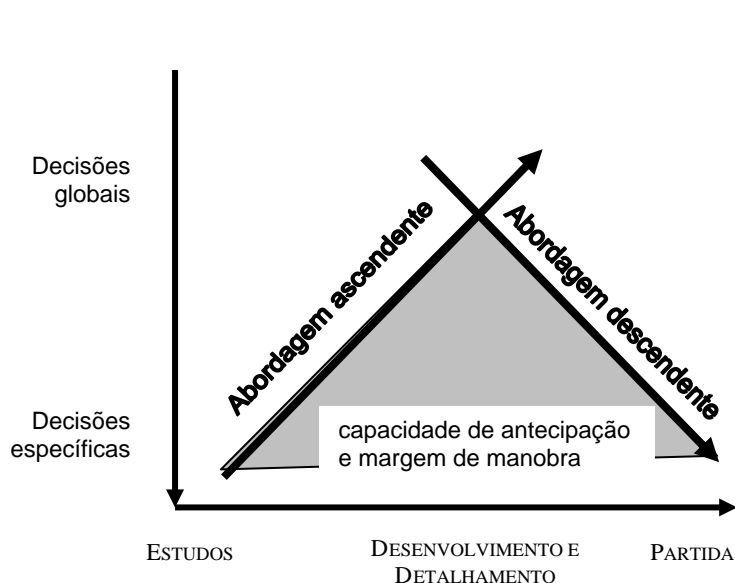


FIGURA 1: Articulação entre as abordagens descendentes e ascendentes. Adaptado de Duarte, 2000.

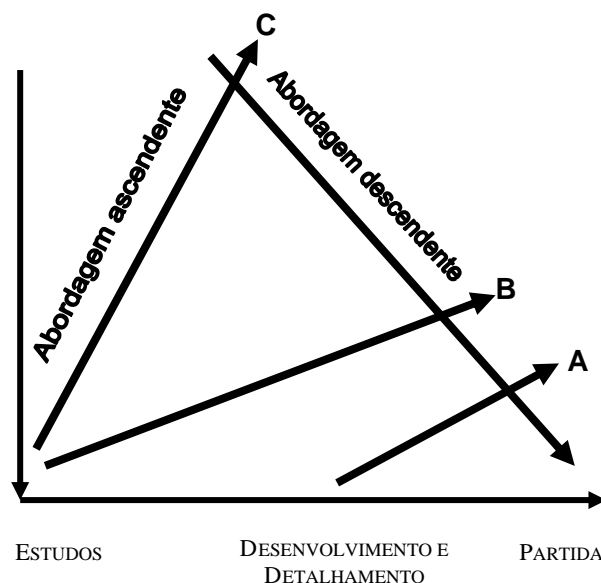


FIGURA 2: Diferentes articulações entre abordagens descendentes e ascendentes. Adaptado de Duarte, 2000.

inconvenientes são os mesmos da situação **A**: "*as margens de manobra para mudanças nos projetos são muito reduzidas, pois as principais decisões já foram tomadas*" (DUARTE, 2000).

Finalmente, na abordagem ascendente representada pela **linha C**, torna-se possível a reflexão entre os agentes envolvidos no projeto (gerentes, técnicos, clientes e usuários) a respeito de suas opções principais. "*Essa combinação das abordagens descendentes e ascendentes permite a descrição e compreensão das inter-relações entre os diferentes componentes do projeto, ampliando a capacidade de antecipação e reduzindo, ao longo do processo de concepção, as incertezas relativas à eficácia do funcionamento futuro.*" (DUARTE, 2000) Assim, gere-se melhor os efeitos de irreversibilidade implicados em qualquer decisão.

Em projetos de sistemas de automação, as dificuldades encontradas na definição dos requisitos, e que geram retrabalho, foram determinadas por causas agrupadas em cinco itens: 1) a própria metodologia de especificação; 2) a escolha dos responsáveis pela definição dos requisitos; 3) a metodologia de explicitação dos requisitos; 4) as dificuldades de comunicação entre especialistas e leigos, e 5) a inversão dos papéis entre fornecedor e cliente, que passa a definir soluções técnicas específicas de automação.

3. O difícil processo de desenvolvimento de sistemas informatizados

A elaboração da especificação funcional comporta uma fase de levantamento de requisitos, a partir de informações fornecidas pelos clientes, incluindo as necessidades operacionais e as finalidades do sistema de controle, base para se definir escopo do projeto. O escopo contém a descrição das funcionalidades e da macroestrutura do sistema, fornecendo as diretrizes para os programadores. Quando os requisitos não são bem definidos, seja porque estão desconectados da atividade dos usuários finais ou porque estão incompletos, o retrabalho será inevitável. Os conceitos de métodos ascendentes e descendentes nos permitirão compreender porque a definição de requisitos apresenta dificuldades.

3.1. A natureza da metodologia de especificação

Os analistas acreditam que o cliente sabe, de antemão, as funcionalidades que gostaria que o sistema tivesse, mas se esquece de repassar informações importantes. Caso houvesse mais tempo na fase de construção da especificação funcional, teriam condições de fazer com que o cliente se lembrasse de tudo o que seu sistema precisaria conter. *“É importante, na especificação funcional, o cliente dizer tudo o que ele pensa e quer, como vai ser. Porque depois ele vai lembrar só no teste de plataforma [teste com simulação do sistema], ou pior, no start-up, e aí o impacto que isso pode gerar é enorme, um retrabalho enorme. Então, tem que gastar tempo na especificação funcional para tentar definir bem como o sistema vai ser, mandar pra ele ver e só começar a fazer quando ele estiver de acordo”* (programador).

Os estudos sobre metodologia de projetos no campo da Engenharia de *software* (BECK, 2000; OLIVEIRA, 2003) já reconheceram ser impossível o usuário saber o que vai querer sem conhecer o sistema, pois a percepção que os usuários têm de suas necessidades: *“também evolui à medida que eles conhecem o sistema. É difícil compreender o valor de uma determinada funcionalidade até que ela seja efetivamente usada, principalmente porque não se pode requerer de um usuário comum a mesma capacidade de abstração que um desenvolvedor possui ao olhar um conjunto de requisitos”* (OLIVEIRA, 2003, p. 16).

A falta de tempo é considerada o maior empecilho para a obtenção de informações que evitariam o retrabalho. No entanto, esse não é o principal motivo pelo qual as informações extraídas do cliente não condizem com sua real necessidade e interesse. Mesmo se o analista ou programador tivesse o “tempo ideal” para extrair do cliente suas necessidades, ele ainda teria dificuldades em obter todas as informações necessárias, pois não se trata propriamente de “esquecimento” do cliente. O motivo real da incompletude das informações decorre de um paradoxo mais fundamental: o cliente não sabe o que pedir para programar o sistema de automação antes de conhecer como os softwares de controle funcionam e o que eles podem oferecer. As necessidades não são definidas (nem definíveis) *a priori*, mas vão-se desenhando ao longo do projeto. Donde a necessidade de uma abordagem que rompa com a temporalidade linear do projeto organizado em fases sequenciais (ver, por exemplo, a metodologia do «objetos intermediários» (CAMPOS, 2002; VINCK, 1999), que aperfeiçoa as simulações e modelos convencionais).

Uma característica fundamental da metodologia de concepção ascendente é aproximação das etapas de concepção e de operação, propiciando uma maior interação entre os profissionais da concepção e da operação, como ocorre nos projetos na “produção enxuta”: *“as equipes de projeto japonesas permanecem no local até bem depois da implantação e operam contínuas mudanças, acumulando, assim, novos conhecimentos, que serão transferidos aos sistemas através das modificações que fazem”* (LOJKINE, 1995, p. 248). Tem-se, assim, um modelo não-linear, que enfatiza a retroação de fases posteriores sobre as fases anteriores e interação concepção/produção.

3.2. Os definidores dos requisitos: quem é o «cliente» afinal?

Um dos principais determinantes do retrabalho é a seleção do responsável pela definição dos requisitos. Os estudos da inteligência artificial sugerem a participação de vários especialistas utilizando técnicas de extração do saber como *brainstorming*, tomada de decisão consensual, técnica de grupo nominal e sessões de exercícios no computador (sobre as técnicas, ver MCGRAW & HARBISON-BRIGGS, 1989). No entanto, as metodologias usadas no desenvolvimento de sistemas especialistas não têm sido eficazes para superar as dificuldades da explicitação do saber prático.

Normalmente, na empresa-cliente, a definição de requisitos não é feita com a participação dos

usuários finais. Na fase de levantamento de dados, o primeiro contato ocorre entre o analista e gerentes, quando se define como será o sistema (equipamentos, grau de automação, funcionalidades, malhas de controle, tipo de interfaces etc.). Em função da escolha restrita dos profissionais responsáveis pela definição dos requisitos do sistema, muita informação pertinente e necessária sobre o processo fica ausente na especificação. Isso ocorre devido ao limite de conhecimento desses profissionais, que desconhecem uma série de informações específicas do processo produtivo. *“Quem faz o levantamento de dados nem sempre consegue todas as informações necessárias. Quando chega lá, fica sabendo de muita coisa, mas há muita coisa também que ele não tem acesso porque quem passa as informações não sabe mesmo, são detalhes que ele não conhece, coisa muito específica da operação”* (coordenador de projetos de automação).

Ora, o saber prático sobre a planta, o processo, os controles realizados pelos operadores ou mesmo pelo antigo sistema, eventualmente a ser substituído, não se restringe a poucas pessoas. Esse saber encontra-se distribuído entre vários profissionais que possuem vivências diferentes, de acordo com a experiência e com a prática diária (BAINBRIDGE, 1999, p. 219). *“Às vezes, a produção só sabe, conhece o limite do processo, mas não conhece o limite dos equipamentos que tem lá. Então, geralmente, quem vai dar esses limites é o encarregado da manutenção”* (programador). Mesmo o usuário final do sistema, o operador da sala de controle, raras vezes é consultado: *a pessoa que vai fazer essa parte, nem sempre envolve o operador. Isso é um erro, o operador é uma peça importante. Deve-se envolver o operador, é ele que vai operar a planta, é ele que vai dar o retorno do que está ruim, do que está bom (...) Normalmente, a gente não entra em contato com o operador antes da plataforma, a não ser num caso que não é o normal”* (programador).

Confundir as categorias de «cliente» e de «usuário» (que são diversos) é um problema real. Entretanto, os programadores, quando têm contato com os operadores, não usam uma metodologia adequada para conhecer a atividade real, baseando-se unicamente no discurso dos operadores, meio limitado para obter informações sobre as necessidades reais de controle do processo. As metodologias ascendentes combinam a presença do operador desde o início da especificação com técnicas apropriadas de observação e entrevista.

3.3. Metodologia de explicitação de regras práticas

A literatura sobre Inteligência Artificial reconhece as dificuldades de explicitação do saber do especialista. *“É sempre difícil para o especialista humano descrever o saber em termos precisos, completos e suficientemente coerentes para poder ser utilizado em um programa de computador. Esta dificuldade, de natureza inerente ao saber, é que constitui a competência humana: ele é sempre subconsciente e pode ser aproximativo, incompleto e incoerente”* (BUCHANAN et al (1983), apud COLLINS, 1992). Com o aumento da competência do sujeito, torna-se cada vez mais difícil a explicitação do seu saber tal como é efetivamente utilizado na ação (MCGRAW & HARBISON-BRIGGS, 1989; WATERMAN, 1986). À medida que um indivíduo vai-se tornando especialista em uma área, não age mais a partir de regras claras e pensamento deliberado: durante sua ação, na maior parte do tempo, ele se orienta pela intuição. Intuição é diferente de recordar experiências da infância ou qualquer processo inconsciente, está relacionado ao agir em dadas circunstâncias, sem recorrer a um pensamento consciente e sistemático, adquirido a partir do acúmulo de vivências anteriores (DREYFUS & DREYFUS, 1986; WATERMAN, 1986). Assim, é pouco eficiente, na aquisição de saber, solicitar que o especialista formule, ele próprio, as regras e os métodos de resolução de problemas. A tendência do especialista é relatar suas conclusões e razões em termos gerais, como se adotasse procedimentos simples.

As técnicas convencionais de entrevista não são suficientes: o que o sujeito diz pode não ser, necessariamente, aquilo que faz. O relato verbal deve ser complementado por dados de observação, como detalhes do comportamento e do contexto específico, além de dúvidas, hesitações, incertezas, comportamento “não-oficial” e também casos nos quais o operador decide não intervir (BAINBRIDGE, 1999). O saber depende do contexto em que se desenrola a ação, inclusive da interação com os outros. Por isso, os informantes não se “lembram” de falar de certos detalhes aos analistas, uma vez que a memória também depende das situações.

3.4. «Diálogo de surdos»: por que especialistas e leigos não se entendem?

Uma outra razão da dificuldade de explicitação do saber, que não foi analisada pela literatura técnica sobre a explicitação do saber, é a diferença de linguagem existente entre sujeitos pertencentes a «mundos» diferentes. O encontro entre essas diferentes linguagens produz verdadeiros “diálogos de surdos”, em que a informação produzida não tem relação alguma com a pergunta feita. O diálogo abaixo exemplifica esta dificuldade de comunicação entre os programadores e os operadores, quando o programador tenta obter uma regra do operador a partir de uma situação hipotética:

Programador: “*O que fazer quando tiver falha de comunicação [entre campo e sala de controle]?*”

Operador: [não responde]

P: *Se tiver falha de comunicação, desarma as guilhotinas?*

O: *Na perda de comunicação, deixa para a operação decidir.*

P: *Mas, se perder a CPU, não consegue operar no campo. Uma CPU sai da rede, não o supervisorio.”*

O: *O que pode acontecer fisicamente?*

P: *Informação mentirosa na tela se a comunicação cair. Se tiver falha de comunicação, não estou lendo a primeira CPU, vai lá em campo partir?*

O: *Não entendi.*

P: [faz um desenho explicando a falha de comunicação]. *Tem uma CPU aqui [faz um quadrado] que comunica com esta [segundo quadrado]. Na hora que esta [a primeira] cai, a outra não sabe que ela caiu. Então, deve deixar o intertravamento atuar ou coloca que o operador só pode abrir no manual? Lá no campo?*

O: [não responde]

P: [faz uma analogia] *Eu sou a CPU1. Você é a CPU2. Eu comunico com o campo. Tudo o que acontecer lá, eu fico sabendo. Eu comunico também com você. Tudo que eu fico sabendo, eu falo para você. Aí eu perco o contato com o campo, não sei mais o que acontece lá. Só que eu continuo acordado, eu não desligo, e falando com você. Só que você não sabe que eu perdi o contato com o campo, você acha que eu estou te passando informação correta. Então, o que fazer? Desliga os exaustores ou deixa ligado? Você não sabe o que está acontecendo no campo, e nem sabe que eu perdi o contato com o campo. Por isso que não tem jeito da operação decidir, entendeu? O operador fica cego.*

O: [não responde].

P: *Vou mudar a pergunta: o que é pior? Deixar o intertravamento atuar ou deixar funcionando?*

O: *Na falha de comunicação, deixa funcionando.*

P *Sério? Nós já fizemos este intertravamento.*

No decorrer do diálogo, percebe-se que o operador não compreendeu o que o programador queria dizer por «falha de comunicação», nem como ela acontecia ou suas conseqüências. O programador precisou explicar-lhe de várias formas: desenhos, analogias, mímicas..., formula a pergunta de várias maneiras, até conseguir uma resposta considerada razoável. No entanto, é

possível, ao final de tudo isso, o operador ainda não ter compreendido muito bem qual era o impacto daquele problema sobre a operação, pois acaba dando uma resposta genérica, certamente possível de acontecer, mas acompanhada de uma expressão facial de dúvida e usando uma tonalidade da voz que deixa transparecer insegurança.

Como mostra o diálogo, o programador, diante da resposta inadequada, pára de perguntar ao operador como deveria ser a regra, e passa a lhe explicar como aconteceria a falha de comunicação na operação. Até esse momento, o programador utilizava a lógica formal com perguntas do tipo «se... então...». Esse tipo de pergunta é típico do diálogo entre os analistas de sistemas e os usuários. Os programadores comunicam-se com o operador segundo a lógica da automação, com perguntas abstratas e descontextualizadas. Não explicitam o saber dos operadores segundo a lógica de utilização ou de operação dos sistemas, para, depois, converter o saber prático em regras formais. Ao contrário, solicitavam do operador um esforço de abstração e generalização do seu saber situado e específico, tarefa que deve ser do próprio analista.

3.5. Definição das soluções pelos clientes

Finalmente, um problema importante no levantamento de requisitos é a inversão de papéis entre informáticos e clientes. Muitas vezes, ao invés dos clientes explicitarem as suas necessidades, já apresentam as soluções que devem constar no software, principalmente quando o cliente possui conhecimentos na área da informática. Essa troca de papéis preocupa alguns analistas: *“Outra coisa que eu insisto com os desenvolvedores [programadores] é sobre as soluções do cliente. Se a gente é a empresa especialista em automação, é a gente que deve saber qual a melhor solução para a necessidade do cliente, qual a melhor tecnologia, o melhor controle, a melhor maneira de fazer”* (programador experiente).

A literatura sobre sistemas especialistas tem deixado claro que, na relação com o cliente, a função do programador é explicitar e formalizar o saber do especialista. Sua função é dar a solução de automação, traduzindo o saber explicitado, normalmente desarticulado, em regras formais organizadas de modo coerente, conforme a lógica da automação. Como os clientes não são especialistas na construção de softwares, caso contrário, não se justificaria a contratação, a um custo elevado, de uma equipe especializada para desenvolver o projeto, as soluções que oferecem nem sempre são as mais adequadas, comprometendo a qualidade do sistema e gerando retrabalho.

Desse modo, é preciso oferecer condições para os analistas realizarem a explicitação do saber dos operadores com eficiência e, a partir daí, construir as regras formais de controle. Os analistas devem obter, em um primeiro momento, conhecimento da operação do processo, conduzindo as entrevistas na perspectiva do operador, para, em seguida, formalizar o saber casuístico em regras gerais. O projeto final do sistema é o encontro de duas lógicas diferentes: uma ascendente - a do operador - que funciona no caso a caso, que o orienta para controlar o processo em situações específicas, às vezes inéditas; e a do analista, descendente, que deve criar um sistema capaz de funcionar de modo automático a maior parte do tempo possível, portanto, submeter várias situações a uma regra geral.

4. Conclusão

A solução dos problemas relatados aqui é fundamental para evitar o retrabalho decorrente de informações lacunares ou inadequadas, obtidas durante o projeto de controle automático. Podemos resumir o princípio geral de uma abordagem alternativa na gestão da

irreversibilidade das decisões tomadas ao longo do PDP. A possibilidade de (re)definir o escopo depende das diversas combinações entre estratégias de projeto descendentes e ascendentes, onde se criam condições de antecipação e de correção, minimizando o retrabalho. Isto implica uma nova temporalidade do PDP, gastando-se mais tempo na especificação, para se economizar no retrabalho e no *start-up*, e a realimentação mais eficaz dos momentos de decisão com informações de campo (métodos ascendentes). Isto é possível, em parte, com o aperfeiçoamento dos métodos de obtenção e validação das informações de campo e dos testes intermediários (testes de plataforma, objetos intermediários...).

O PDP é uma criação coletiva. No entanto, a mistura de papéis pode ser prejudicial, transformando o fornecedor de softwares em um mero executante. A gestão dessa relação é complexa precisamente porque exige uma definição precisa dos papéis de cada agente, assim como do conhecimento especializado que cada um possui enquanto especialista em um certo campo, ao mesmo tempo em que esses diversos especialistas devem cooperar entre si, levando em conta o ponto de vista do outro e traduzindo suas necessidades em exigências de projeto.

Propostas específicas sobre uma nova abordagem foram detalhadas em Ferreira (2004). Ressaltamos, aqui, apenas o princípio geral: as especificações técnicas devem se inserir em um «contrato social» mais amplo, cujo objetivo é definir as condições de relacionamento dos agentes, em especial entre cliente e fornecedor. Nesse sentido, nossas propostas consistem em uma série de instrumentos metodológicos que favoreçam a confrontação de lógicas diferentes (equipes multifuncionais, participação da produção e da manutenção, uso de objetos intermediários diversos...), de modo a manter o mais aberta possível, o máximo de tempo possível, a definição dos objetivos e dos requisitos do projeto (TIGER, 1991). Seja pela antecipação, seja pela retroalimentação, o que vai acontecer interfere sobre o já acontecido, tornando o PDP menos determinado por decisões irreversíveis e criando condições mais favoráveis para «se projetar certo da primeira vez».

Referências

- BAINBRIDGE, L. (1999). Verbal reports as evidence of the process operator's knowledge. *Int. Human-Computer Studies*, n.51, p.213-238.
- BECK, K. (2000). *Extreme programming explained: embrace change*. Boston: Addison-Wesley.
- CAMPOS, N. (2002). *Equipes multifuncionais de projeto*. Dissertação Mestrado. UFMG, Belo Horizonte.
- CARVALHO, L.C. (1988). *Análise de sistemas*. Rio de Janeiro: Livros Técnicos e Científicos.
- COLLINS, H.M. (1992). *Artificials experts*. Seuil: MIT Press.
- DREYFUS, H.; DREYFUS, S (1986). *Mind over machine*. New York: Free Press.
- DUARTE, F. (2000). Complementaridade entre ergonomia e engenharia em projetos industriais. IN: DUARTE, F. (org.) *Ergonomia e projeto na indústria de processo contínuo*. Rio de Janeiro, Lucerna, pp. 11-21.
- FERREIRA, R.B. (2004) *Diálogo de surdos: a difícil explicitação do saber entre programadores de software e operadores de fábrica*. Dissertação de mestrado pela Engenharia de Produção: UFMG.
- LOJKINE, J. (1996). *A revolução informacional*. São Paulo: Cortez.
- MCGRAW, K.L.; HARBISON-BRIGGS, K. (1989). *Knowledge acquisition*. Englewood Cliffs: Prentice Hall.
- SILVA, C.; LIMA, F. (2000). A objetivação do saber prático em sistemas especialistas. In: DUARTE, *op. cit.*
- TIGER, H. (1991). L'établissement du Cahier des Charges des équipements automatiques. *Colloque A.R.R.P.*, MRT, Janvier 1991.
- VINCK, D. (1999). *Ingénieurs au quotidien*. Grenoble, PUG.
- WHEELWRIGHT, S. C. & CLARK, K. B. (1992). *Revolutioning product development*. New York, Free Press.