

Ambiente de Desenvolvimento de Software - ADS e a Gerência de Informática. Por que não é fácil adotar ferramenta CASE em uma Empresa?  
Antônio Carlos de A. Rizzo  
Arndt Von Siza

RESUMO: O trabalho apresenta uma proposta visando o estabelecimento de um Ambiente de Desenvolvimento de Software - ADS. A proposta aborda o ADS de forma completa considerando tanto a integração entre as diversas fases como o atendimento a todas as funções gerenciais.

PALAVRAS-CHAVES: Ambiente de desenvolvimento de sistemas, Ferramentas, CASE, Produtividade, Gerência do desenvolvimento, Processo de desenvolvimento

#### INTRODUÇÃO

A demanda no desenvolvimento de softwares é grande e crescente. Cada vez mais novos segmentos se incluem como usuários da Informática pressionando por mais e melhores softwares [1]. A sociedade se informatiza e a indústria de equipamentos se apresenta num caminho seguro no sentido da maturidade, padronização e produção organizada.

A área de software não alcançou ainda esse patamar. Há ainda indefinições, falta de padronização, de eficiência, de qualidade, de um processo de produção bem definido. Estes aspectos negativos se manifestam aos usuários que precisam de software através de duas características indesejáveis, não raro, insusceptíveis que são tempo e custo de desenvolvimento.

#### Medidas visa

Medidas visa não minimização de seus efeitos, obrigam a consideração de um grande número de métodos e técnicas tais como metodologia de desenvolvimento, ciclos de vida, técnicas de desenvolvimento, ferramentas de apoio, estimativa de custos e prazos e planejamento e controle do desenvolvimento. Estes, dentre muitos, são fatores que precisam ser estudados e tratados num Ambiente de Desenvolvimento de Software. O exame de cada um deles e a formulação de uma proposta para seu encaminhamento, por si só, representa um grande trabalho e exige tempo e recursos consideráveis. Um agravante natural é que não basta a solução de cada elemento. É mandatória a integração harmonizada de todas as propostas desenhando um ambiente viável, considerados todos os seus componentes, todas as suas manifestações.

A Informática tem evoluído rapidamente. Métodos novos de desenvolvimento surgido de várias fontes e aplicáveis aos vários casos e tipos de software. Mas como adaptá-los? Porque não é fácil adotar ferramenta CASE em uma empresa?

Todos são unânimes quanto a necessidade de que se adotem ferramentas de produtividade. No entanto, isoladamente tem se mostrado de difícil trato e ainda

são poucos os casos de adoção institucional de um ambiente automatizado. Nosso pensamento é que isso se deve a 2 aspectos principais:

- Falta de integração entre todas as fases do desenvolvimento
- Falta de apoio gerencial

Destacamos a importância do aspecto gerencial. Um moderno AD5 tem que viabilizar a gerência do desenvolvimento do software. Nesta área tal vez esteja o principal desafio uma vez que as ideias e propostas não estão consolidadas.

Na área técnica, no que toca ao apoio, diversas fases e mesmo quanto a integração, certamente se dá o maior defasamento entre o que se apresenta nas Academias, e disponível através de literatura, e o que se pratica na Indústria de Software. Orega a ser de 10 anos essa distância [2]. Tempo exageradamente grande para qualquer segmento mas surpreendente no caso de uma área como a Informática que não se cansa de obsoletar, sempre em espaço curto de tempo, suas próprias novidades.

A adoção de técnicas e métodos novos nos setores de produção de softwares exige uma abordagem abrangente que considere todos os fatores do ambiente, incluindo o aspecto gerencial, com forte apoio em ferramentas mecanizadas. Orega a ser surpreendente o fato de ser a função Informática uma das últimas a se valer das vantagens do uso dos recursos de Informática. Paradoxal (!?!): casa de ferro e espelho de pau; Síndrome do filho do sapateiro, etc.

Não pode mais ser assim. Atualmente os analistas se deparam com o desenvolvimento de software cada vez maiores e mais complexos. Nesses desenvolvimentos estão envolvidos técnicos de várias especialidades tratados, certamente na maioria dos casos, com funções on-line e de tempo real.

Os altos custos do desenvolvimento de software, a recorrência da baixa produtividade das equipes de desenvolvimento e a complexidade e portabilidade dos problemas a serem tratados exigem que se considere os instrumentos mais potentes no encaminhamento de suas soluções.

#### PROBLEMAS ENFRENTADOS NO DESENVOLVIMENTO DE SISTEMAS

Podemos observar, dentre outros, os seguintes aspectos negativos presentes nos ambientes atualmente existentes nas empresas no que toca a desenvolvimento de software.

##### - GRANDE BACKLOG

A velocidade do crescimento da demanda tem se mostrado muito maior do que a capacidade das áreas de desenvolvimento em gerar produtos e a tendência. Esta fila de espera acaba por ser causa e causadora de ambientes inadequados.

Por um lado decorre da inadequação do ambiente, que por estabelecer uma velocidade morosa, aumenta seu conteúdo. Por outro lado é causadora uma vez que pressiona no sentido de desenvolvimentos urgentes impedindo, na prática, que se invistam recursos no sentido da modernização dos métodos de trabalho. Uma fila de projetos de 5 anos é inadmissível. - GERÊNCIA INVIABILIZADA

O informalismo e a imprecisão com que o processo é normalmente implementado fazem mortalmente a possibilidade de surgimento de uma solução adequada e

econômica. Os profissionais das áreas usuárias costumam adotar um comportamento extremamente passivo ante a pressuposta "complexidade natural" inerente ao uso de recursos da Informática. Dizemos pressuposta uma vez que aspectos técnicos de processamento eletrônico de dados não deveriam estar presentes quando se busca, junto às áreas usuárias, descober e definir a essência do sistema que se quer desenvolver.

Os Engenheiros de Software por sua vez são, não raro, mal formados e não dispõem de métodos, técnicas e ferramentas capazes de proporcionar condução coerente e clara do processo de desenvolvimento.

O comportamento dos gerentes e engenheiros de software, bem como o processo de desenvolvimento, são impregnados de formalismos, imprecisões e da falta de uma metodologia rigorosa que conduza o processo de desenvolvimento. O engenheiro de software conta quase que exclusivamente com seu talento pessoal tanto no que tange aos aspectos do sistema que se desenvolve como na sua capacidade de influenciar usuários no sentido de aceitar tanto o processo como o produto do desenvolvimento.

Neste contexto planejamos ficar bastante prejudicados. Nesse ambiente, entocampamente o progresso fica bastante prejudicados. Nesse ambiente, desobediência a princípios básicos de engenharia, sem processo formal e previamente bem determinado, a ação gerencial fica, na prática, bastante prejudicada quando não fica inviabilizada.

É com frequência que o "praticamente pronto" e não é incomum um software ficar nesse estado durante mais da metade do seu tempo de desenvolvimento. Os projetos não terminam e os recursos humanos não se desaloçam. A produção torna-se de difícil dimensionamento e gerenciamento. Ao terminar o trabalho de desenvolvimento parte da equipe se desloca para a atividade de manutenção. Não se cumprem os prazos por mais que se sustentem os dados do conclusão dos projetos.

##### - BAIXA QUALIDADE NOS SOFTWARES DESENVOLVIDOS

Os produtos gerados são frequentemente de baixa qualidade. Não tanto no que diz respeito à qualidade do software quanto sua performance nas execuções em computador mas no funcionamento do sistema junto às áreas usuárias. Como o sistema é mais especificado e mal desenvolvido, algumas de suas funções não "cabem" na sistemática dos usuários. Este aspecto é tão grave que não são raros os casos em que não são implantadas todas as funções do sistema por inadequações insuperáveis realidade das áreas usuárias. Mais do que custo isso provoca descuido e obriga a uma complexidade desnecessária tanto na produção quanto na manutenção do software.

##### - MANUTENÇÃO DEMORADA, DIFÍCIL, PREDATÓRIA

Durante as necessárias manutenções do sistema é pago um preço alto por essa baixa qualidade e, uma vez mais o informalismo e a imprecisão obrigam que cada alteração se dê de forma difícil, demorada e, o que é pior, predatória. Não há documentação, faltam padrões e falta tecnologia. O sistema nem está pronto e já são necessárias manutenções para corrigir equívocos de uma especificação que não foi documentada. A manutenção corre num clima de "tapar buraco", desordenadamente, sem registro e, não raro sem análise global criando uma quantidade enorme de problemas e levando a já frágil qualidade do software.

É num ambiente mais ou menos assim que se tem desenvolvido software. Baixa qualidade nos métodos de trabalho, tendo como decorrência baixa produtividade e pouca qualidade nos produtos gerados. Profissionais com formação insuficiente proporcionando soluções impróprias com baixa produtividade. É isso que se pretende mudar quando se discute "Ambientes de Desenvolvimento de Software".

#### CAMINHOS PARA A SOLUÇÃO

O infante processo de desenvolvimento de software necessita investimentos efetivos na busca de produtividade. A medida da produtividade também precisa ser cuidadosa, sempre comprometida com o produto final e não com o processo. Medidas como "linhas corretas de código produzidas" podem se mostrar falaciosas quando se quer medir produtividade do ADS [3]. Tem que ser medido o volume do software efetivamente produzido, com qualidade e dentro do custo e prazos estabelecidos.

- Três caminhos podem ser eleitos como principais, na busca de solução:
  - mais recursos humanos com competência
  - Automação do processo de desenvolvimento
  - mais produtividade dos recursos humanos

As palavras-chave são competência, automação e produtividade.

Esses três conceitos na verdade precisam convergir e juntos estabelecerem um ADS.

A quantidade de recursos humanos normalmente envolvida no desenvolvimento de um software não pode crescer indefinidamente [4]. O que se precisa aumentar são competência, produtividade e automação.

Métodos de trabalho, linguagens de representação e seus aspectos sintáticos e semânticos precisam ser fixados, junto com processos de do tempo práticos.

Isto feito, um grande esforço de capacitação precisa ser desenvolvido no sentido da elevação do nível de conhecimento em "Linguagens" que deixem fluir e entre os profissionais envolvidos, sejam eles analistas, programadores ou usuários, modelos que possam significar por si, conhecimento registrado e entendido por todos.

Capacitação, aplicada nessas bases, faria surgir competência e valores técnicos de conhecimento, tanto técnicos como práticos, substituiriam o usual talento impregnado de informalismo que não cria nenhuma oportunidade de medida efetiva de progresso ou produtividade.

Automação é irreversível. Quanto mais a atividade de Engenharia de Software se

formaliza, quanto mais se fixem aspectos metodológicos, mais concretamente se poderão ser utilizados os recursos da automação. A dificuldade atualmente inexistente de se utilizar as técnicas.

Produtividade é conta de resultado. Fixadas as técnicas e métodos, alcançada competência nas suas práticas, automatizadas as partes adequadas dos processos de obtenção e estabelecidos critérios práticos de acompanhar progresso e qualidade, a produtividade será mais facilmente obtida e acompanhada através de métricas mais efetivas, qual seja, o volume de produtos com qualidade assegurada produzidos por unidade de tempo.

Os caminhos propostos não são fáceis de serem trilhados.

Capacitação de mão-de-obra requer muito tempo; a rigor ainda temos que e formalizar em que, completa e exatamente, precisa nos capacitar.

Automação carece de tecnologias a serem desenvolvidas.

O processo de busca da melhor forma de fazer sempre existirá, é permanente. A dificuldade não deve ser descorajadora. Ela é combatível com o port e do problema que se almeja resolver. Todavia ainda temos que adicionar a esses 3 ingredientes o aspecto gerencial de planejamento e controle do trabalho.

## AMBIENTE DE DESENVOLVIMENTO DE SOFTWARE - ADS

### 1. ENGENHARIA DE SOFTWARE

Todas as questões colocadas passam pela arrumação do ambiente de desenvolvimento. Essa arrumação pressupõe uma mudança de comportamento e a adoção de uma postura engenharia no lugar de uma postura artística ou artesanal.

Por Engenharia de Software deve ser entendido o conjunto de disciplinas técnicas e gerenciais que sistematizam a produção e manutenção de produtos de software com qualidade, dentro de prazos e custos estimados e com progresso controlado.

### 2. O QUE É UM ADS ?

Um ADS é um conjunto integrado de disciplinas para apoio gerencial, estratégias de desenvolvimento (ciclos de vida), métodos, técnicas e ferramentas, que visa a adoção de uma sistematização na produção e manutenção de software.

Pensar em um ambiente de desenvolvimento de sistemas com essas como tags é incorporar os princípios da Engenharia de Software, na maturidade de sua prática, estabelecer as condições básicas para eliminar os aspectos negativos comentados. Todavia é fundamental que saibamos da realidade que tudo isso determina.

A literatura tem tratado do assunto de ambientes de desenvolvimento principalmente através do aprofundamento nos produtos CASE - COMPUTER AIDED SOFTWARE ENGINEERING.

O nome CASE, e suas variações, tem sido empregado para significar vários aspectos e, na maioria dos casos, são mais detalhadamente explorados em seus aspectos de automação visando ferramentas específicas. O aspecto de automação é mandatório. Só através da automação de grande parte do processo se poderá tratar adequadamente ADS.

É imperativo que se invista no sentido de mudar a situação atual onde é intensa a aplicação da mão-de-obra para desempenho de tarefas extremamente laboriosas transformando o processo, constituindo-o de tarefas mais centradas no conhecimento.

Toma-se como um todo, a literatura apresenta 4 tipos de CASE [3], [5]. Aquelas centradas em linguagem de programação que é o caso do ambiente ADA, os orientados para a estrutura que são independentes de linguagem, os do tipo TOOLKIT que proporcionam várias ferramentas e também são independentes de linguagem e, finalmente aquelas abrangentes baseadas em métodos de trabalho.

O que propomos é ADS baseado em métodos de trabalho concentrados em banco de dados e sistemas de banco de conhecimento que estabeleçam o ambiente como um todo e

onde um repertório de ferramentas, mecanizadas ou não, encontram sua oportunidade e adequabilidade quando o engenheiro de software é colocado diante de um problema específico.

Sem dúvida é preciso mais que simplesmente mecanizar tarefas. É necessário tratar o ambiente como um todo e criar uma arquitetura que além de conter ferramentas e forte apoio mecanizado estabeleça métodos que deem coerência e oportunidade a cada atividade do conjunto. Há várias formas de se definir um ADS, mesmo aquelas baseadas em métodos. Por conta disso vamos especificar sucintamente o que entendemos ser um ADS.

### 3. OBJETIVO GERAL DE UM ADS.

O objetivo de um ADS é, diante de solicitações das áreas usuárias e respeitando as restrições e condicionantes existentes, investigar e descrever os requisitos a serem satisfeitos e desenvolver, implantar e manter sistemas para atendimento das necessidades.

O desenvolvimento tem que se dar de forma planejada e o andamento do processo precisa ser controlado e reportado desde os primeiros momentos da manifestação das intenções até a implantação do produto final tanto quanto aos seus custos quanto a qualidade dos produtos parciais.

### 4. OBJETIVOS ESPECÍFICOS DO ADS

Quanto ao gerente

No atendimento ao Gerente do Desenvolvimento reside um fato preponderante não só do sucesso de um ADS como até na viabilização de sua adoção.

A atividade de desenvolvimento de software precisa de um instrumento que encontre a época de sua quase inviabilização gerencial.

Para tanto um ADS precisa conter instrumentos automáticos, e integrados com todas as outras ferramentas técnicas, para dentro outros apoiar:

- Identificação de Projetos
- Escolha de estratégias de desenvolvimento (Ciclos de Vida)
- Estimativas de custo e tempo
- Walk-through
- Alocação de recursos humanos
- Avaliação de custo
- Estabelecimento e manutenção de programa detalhado de trabalho
- Controle das atividades

Quanto ao Engenheiro de Software

O Engenheiro de Software precisa ser atendido na utilização de suas ferramentas.

Integração e entre as ferramentas é pressuposto mandatório, bem como um potente dicionário ou

e descreva cada objeto com significado correto dentro da semântica de cada representação gráfica das ferramentas adotadas.

Neste contexto precisam estar disponíveis no ADS e, insistimos, de forma integrada, ferramentas para apoio, dentre outros, a: Diagrama de Fluxo de Dados, Diagramação de Arquitetura, Diagrama de Estado e Transição, Rede de Petri, Diagrama de Entidade e Relacionamentos, Dicionário de Dados, Diagrama de Objeto, Protótipo, Diagrama de Estrutura de Módulos, Modelagem de Dados, Edição de Telas e Relatórios, Programação "Automática", Auxílio a Testes/Simulação, Gerência de Configurações, Engenharia Reversa e Auditoria de Alterações.

#### QuantoadCliente

Identificar as necessidades e expressá-las em termos de projetos, qualificando recursos e prazos necessários aos seus desenvolvimentos.

- Realizar levantamento de necessidades junto às diversas áreas e suas áreas do cliente e exprimi-las em forma de projetos que explorem os requisitos e restrições.

- Desenvolver os projetos selecionados pelo cliente com participação efetiva de elementos das áreas usuárias mantendo todos informados do programa de trabalho, pendências, dificuldades e mudanças de rumo.

- Identificar alterações necessárias nos sistemas existentes sejam elas motivadas por imperativos ou ditadas por novas necessidades mandatórias e prementes. Novas necessidades não mandatórias ou prementes devem ser tratadas como novos projetos.

- Desenvolver normas e procedimentos que determinem o novo comportamento do usuário e que estejam completamente alinhados com as aquisições de dados e a produção de saídas do software que eventualmente esteja sendo implantado.

- Implantar, treinar e apoiar todos os elementos das áreas usuárias na utilização das normas e procedimentos implantados bem como nas alterações eventualmente promovidas.

- Administrar um processo para apoiar o estabelecimento de prioridades para o desenvolvimento de projetos.
- Administrar um processo que garanta um fluxo de informação para o cliente, ao longo de todo o desenvolvimento, sobre todos os aspectos do seu andamento, privilegiando aqueles voltados para aplicação dos recursos e desvios, no tempo e nos propósitos do projeto.

Para aplicação dos recursos e desvios, no tempo e nos propósitos do projeto, identificar fatores que influenciam na sua qualidade.

- Dentre outros devem ser considerados tarefas programadas e não realizadas, tarefas não programadas realizadas, pendências, tarefas em atraso, tarefas antecipadas, próximos produtos a serem gerados, resultados das Revisões de Revisão, Projetos concluídos, Projetos em andamento, Recursos alocados, Próximas liberações de recursos, Projetos previstos, Novos projetos identificados no período, etc.

#### QuantoadCPD

Um ADS, conforme colocado neste trabalho, por pretender atender a empresas de médio e grande porte quanto ao uso dos recursos de Informática, precisa considerar o tratamento do CPD. O CPD é aqui colocado como a área que detém e administra o recurso computacional central, aí considerado o main-frame, a gerência da rede, a produção do Batch, etc. Esta função tem que ser apoiada pelo ADS. De lo menos, através das seguintes atividades:

- Implantar, treinar e apoiar pessoal da produção para a execução periódica do software visando seu adequado funcionamento tanto no que diz respeito ao atendimento dos seus usuários como no que diz respeito às condições de recursos computacionais (Hardware, software básico, software de apoio).
- Apoiar os usuários finais nas suas atividades de desenvolvimento, produção e manutenção de Aplicações Computacionais.
- Promover alterações necessárias no software, visando, além de atender as condições do item anterior, melhorar performance e atender requisitos novos dos usuários.

#### 5. MACRO FUNÇÕES DO ADS

Um Ambiente de Desenvolvimento de Software possui basicamente 3 grandes funções:

- Planejamento e Controle
- Desenvolvimento de Projeto
- Manutenção de Software

Cada uma dessas grandes funções é um mundo imenso de complexidades e complexidades gerenciais, metodológicas, técnicas, tecnológicas e administrativas. Todavia estas grandes funções cobrem todo o espectro de desenvolvimento de Software qualquer que seja o ângulo pelo qual se queira analisá-lo. Cada uma dessas grandes funções possui seus propósitos, processos e produtos que precisam estar harmonicamente estabelecidos para constituírem um ADS. O diagrama da figura 1 apresenta o ADS com suas 3 macro funções constituintes.

#### 6. O PLANEJAMENTO E CONTROLE

A função Planejamento e Controle organiza todo o trabalho do ADS e controla o andamento de todas as suas atividades identificando as diversas necessidades para os diversos projetos.

No contexto interno do ADS a função Planejamento e Controle determina a estratégia para o desenvolvimento de cada projeto preparando programa detalhado do trabalho a ser seguido em cada um estabelecendo claramente o produto final, os produtos parciais e os pontos em que o desenvolvimento sofrerá Controle de Qualidade.

Com base nos programas detalhados de trabalho esta função exerce controle sobre o andamento de todos os projetos e, além de manter todos as áreas do Cliente informadas, promove as correções de curso que as circunstâncias recomendarem.

Nesta importante função do ADS são executadas subfunções apresentadas na figura 2.

Para que se comece o desenvolvimento de um software é necessário que se tenha um escopo bem definido e, principalmente se saiba como o sistema a ser desenvolvido se encaixa no conjunto de atividades da empresa. Quando esses aspectos estiverem bem definidos ou se mostrarem sem importância, não é necessário que se proceda a uma Análise Funcional.

Todavia, se estamos tratando da informatização de uma função que se configura corporativa e sistêmica no contexto da organização é mandatório que se faça um estudo visando esclarecer suas relações funcionais. Tome mos como exemplo o Sistema de Recursos Humanos conforme apresentado no gráfico da figura 3.

Se o projeto tem como objetivo desenvolver suporte computacional para apoiar isolado a atividades de qualquer das funções integrantes da área de Recursos Humanos estaremos, possivelmente, diante de tarefa cujo contorno, objetivos e relacionamentos sejam de fácil definição.

Desenvolver um software para apoiar a atividade de Acompanhamento dos Treinamentos dos funcionários ou um software para apoiar a um Processo de Seleção ou para apoiar a Controle de Mão-de-Obra Temporária, cada um destes propósitos tomados independentemente uns dos outros, configuram projetos que, maior parte dos casos, são de simples especificação e desenvolvimento.

Situação de outra magnitude é o desenvolvimento de um projeto que pretenda implantar um Sistema para atender a todo o escopo da área de Recursos Humanos colocada na figura 3. Nesta abrangência não se pode, sob pena de grande prejuízo, iniciar o atendimento a uma qualquer atividade, sem considerar suas relações com todas as outras. A idéia, por exemplo, de centralizar o armazenamento de dados em um único acervo, obriga a que todos tenham um mesmo entendimento do significado de cada dado armazenado e tornado disponível para a organização.

Isso estabelece uma "linguagem" no tratamento de todos os aspectos da função. Diga que se identifique um único e mais adequado ponto para aquisição e manutenção de cada dado, e este ponto nem sempre é beneficiado por este esforço. É mandatório que prevaleça a idéia de sinergia. Fica evidente que descrevemos duas situações diferentes:

- desenvolver um software para apoiar a uma atividade de uma função operativa de forma isolada e independente - desenvolver um sistema corporativo estabelecendo uma solução para todo o contexto de uma função de controle da Organização

É no tratamento do segundo caso que se torna necessária uma Análise Funcional cujo objetivo é dominar a complexidade do ambiente estabelecendo uma arquitetura que aborde todos as atividades dentro do contexto e proporcione um planejamento do trabalho através da segmentação da solução em Projetos de Desenvolvimento de Sistemas.

Nunca haverá recursos para desenvolver tudo e o estabelecimento de prioridades é questão muito carregada de aspectos políticos dentro das empresas. Esta função pode ser apoiada por instrumento do ADS que classifique os projetos, pontuando os segundo parâmetros e critérios estabelecidos e ponderados; alguns exemplos de parâmetros são Inovação tecnológica, Maturidade do usuário, Maturidade dos...

riedade dos analistas. Nível de mudança que provoca. Acervo de dados que adquirir e manter. Nível Estratégico para a empresa, Exigência legal, Oportunidade política interna, etc.

Quem não sabe onde quer chegar não precisa escolher um caminho. Quem sabe onde quer chegar tem que procurar, a priori, estabelecer uma forma de alcançar seu objetivo, planejar o seu trabalho.

Em projetos de Informática, dentre outros, isso se caracteriza por um Programa Detalhado de Trabalho. Sua falta dá lugar ao surgimento da síndrome do "praticamente pronto". Não são raros os casos de projetos que passaram no status de "praticamente pronto" mais de metade do tempo do seu desenvolvimento sem que pudesse haver uma efetiva ação gerencial por não ser fácil, e às vezes não ser possível, se realizar uma análise do que realmente falta fazer.

Só a construção de um cuidadoso programa de trabalho resolve este problema [6]. Neste Programa de Trabalho devem ser listados todas as tarefas necessárias ao alcance do objetivo. Quanto mais detalhado for o Programa de Trabalho, melhor será seu desenvolvimento e seu acompanhamento.

Um Programa de Trabalho bem elaborado, além dos benefícios que proporciona como instrumento de planejamento e controle é um elemento imperdível para a organização do trabalho, esclarecimento a cada pessoa envolvida de sua exata participação no projeto e viabilizador do trabalho com junto e partilhado harmonicamente.

Assim colocado o programa de trabalho permite análise de impactos e efeitos no processo de desenvolvimento de software provocado por imprevisões e atrasos, principalmente nas atividades iniciais do projeto.

Frederic Brooks [4] nos ensina que um projeto não atrasa um ano de uma vez; ele atrasa um dia de cada vez, 365 vezes. Só um Programa de Trabalho bem elaborado proporcionará ajuste fino em todo o processo de desenvolvimento. Para que se possa elaborar um programa detalhado de trabalho é mandatório que se conheça bem o que se pretende desenvolver e se tenha escolhido uma estratégia adequada para fazê-lo.

Como decorrência da identificação do Projeto e da análise de seus objetivos, escopo e características será estabelecida uma Estratégia para seu desenvolvimento. As etapas pelas quais passa um projeto de sistema não podem ser rigidamente fixadas. Todas as pretensões terão que passar por uma fase de identificação, que estabelece sua abertura enquanto projeto, e a fase de implantação do Sistema, que, de alguma forma, estabelece sua conclusão. Todavia para seu desenvolvimento há que ter várias alternativas.

Um Sistema Corporativo, abrangente em termos de escopo, profundo a nível de comportamento de atividades da empresa, inovador em termos tecnológicos, sensível em termos políticos, etc, não pode ser desenvolvido sob o mesmo ciclo

de vida que um projeto tipo "tê-1 mpor lme". As estratégias de desenvolvimento, as atividades necessárias a construção do sistema são diferentes. O quadro colocado na figura 4, a título de exemplo, apresenta 4 possíveis estratégias para desenvolvimento.

Na alternativa 1 está sendo considerado o desenvolvimento de um projeto complexo. É pressuposto um Sistema de grande porte e todas as fases do ciclo de vida proposto precisam ser desenvolvidas para domínio da complexidade e elaboração de uma adequada solução.

Na alternativa 2, já é entendido que a área a ser estudada dentro do escopo do projeto não oferece maiores complexidades não sendo portanto necessária uma etapa de planejamento proporcionado pela Análise Funcional podendo partir da identificação para o desenvolvimento do projeto.

Na alternativa 3 é pressuposto maior simplicidade ainda sendo considerado que análise de sistemas e construção do projeto computacional podem se dar no âmbito de um só trabalho para construção do sistema.

Na alternativa 4 é pressuposto que o problema encontra ótima solução com a elaboração de um relatório ou uma tela e que sequer é necessário a aquisição de novos dados nos colocando assim diante da possibilidade mais simples.

Estas considerações não esgotam o assunto nem pretendem estabelecer uma metodologia. O que precisa ser entendido é que antes de iniciarmos o desenvolvimento de um projeto precisamos nos certificar que está claro para todos o que temos efetivamente desenvolver e como vamos fazê-lo. Se sabemos isso, podemos estabelecer uma estratégia e programar detalhadamente o trabalho. Se não sabemos isso, ou fazemos uma Análise Funcional para apoio das decisões, ou iniciamos o desenvolvimento numa postura de protótipo.

O Controle do Desenvolvimento é antes de tudo dependente de uma postura gerencial que provoque, no corpo técnico, um comportamento voltado para esta preocupação. O Controle não pode ser visto nem usado como uma postura política. Junto ao corpo técnico. Precisa ser encarado como a única forma de aprofundamento de todos no processo de desenvolvimento, compromisso entre analistas e usuários e viabilizador de uma análise com clareza, realista e constante do onde estamos, o que efetivamente fizemos e o que falta fazer. Quando podemos agir o que fazemos e expressá-lo a través de algum número conhecido o que fazemos. Quando isso não ocorre nosso conhecimento é, no mínimo insatisfatório (Hambray 88). O controle do andamento dos projetos tem que ser encarado como elemento de crescimento profissional uma vez que esclarece o efeito de cada problema no processo de desenvolvimento, proporcionando amadurecimento e visão realista da atividade de desenvolvimento de software. Está intrinsicamente ligado ao planejamento e a programação de trabalho que se tenha feito para o projeto. Se o planejamento e o programa de trabalho tiverem sido feitos detalhadamente e rigorosos, o controle poderá ser minucioso. Se ao contrário tiverem sido feitos de maneira superficial e informal o controle terá que se ater ao nível da superficialidade e da informalidade. Diante de um programa de trabalho bem elaborado o controle fica não só viabilizado como fácil, não trabalhoso e funciona como um elemento catalizador das discussões e das análises do andamento.

Em termos de apoio mecanizado, o ADS tem que proporcionar ferramenta, assessoria à ferramenta de programa de trabalho, que analise automaticamente todo o andamento, todos os desvios de curso e os efeitos de atrasos de cada atividade no todo do projeto tanto no que diz respeito a prazos como a custos

e alocação de recursos humanos. Tem que permitir ao gerente análise de sensibilidade do tipo WHAT IF no sentido de mudanças diversas na condução do projeto tanto no que tange a um replanejamento geral como um replanejamento em trechos e seus efeitos no todo. A existência de uma bem estruturada função de planejamento e controle proporciona mais objetividade e maior clareza na função de desenvolvimento de projetos.

O Projeto chega à função Desenvolvimento com escopo e objetivos identificados, mesmo que genericamente, mas, fundamentalmente, com uma estratégia de desenvolvimento clara e precisamente definida. A partir deste ponto o desenvolvimento de sistemas se trata de um aprofundamento no ambiente usuário visando conhecer e dominar suas complexidades para, finalmente, definir as normas e procedimentos que estabeleceram o novo comportamento da área usuária e o software que irá apoiá-la. O objetivo do desenvolvimento de projetos, conforme colocado nesta proposta é, portanto, um estudo profundo no usuário e a elaboração de normas, procedimentos e software que promovam uma evolução no seu comportamento eliminando ou minimizando os problemas colocados identificados.

COMO PODE SER ENCARADO O DESENVOLVIMENTO DE PROJETOS ?  
Dentro deste contexto, o desenvolvimento de projetos, desde os primeiros momentos até a concepção final do sistema, pode ser encarado como um suceder de elaboração de modelos. Nos seus mais diversos níveis de abstração, o trabalho se desenvolve construindo gráficos representativos do estágio de conhecimento, qualquer que sejam a metodologia e a técnica gráfica utilizadas, e especificando os objetos nele contidos. Ao longo dos diversos passos da metodologia ou estratégia adotada, numa aquisição progressiva de conhecimento, o engenheiro de sistemas ou software vai, aprofundadamente, alimentando uma Base de Software. Esta alimentação, a cada fase e do processo, resume-se sempre na concepção de um esquema gráfico e na especificação dos objetos nele contidos. A figura 6 esquematiza esta idéia.

As especificações dos diversos objetos das diversas representações gráficas utilizadas pelo modelador vão compondo o dicionário de dados da organização. O dicionário de dados, desta forma construído, e constitui no agente integrador de todas as representações estabelecendo, de forma definitiva e única, a semântica de cada objeto. Estas especificações, que a

princípio são do Sistema, na maturidade do sistema desenvolvido e implantado, passam a pertencer a organização com um todo estando disponível para quem delas necessitar podendo compor outras representações gráficas de outros sistemas. Desta forma, a criação e armazenamento do conhecimento progressivamente adquirido se dá independente da representação gráfica e da metodologia utilizadas. O permanente e integrador é o dicionário com suas especificações. A cada instante é fundamental que haja uma metodologia e que cada uma de suas fases tenha uma forma estabelecida de ser desenvolvida. Todavia o ADS tem que ser aberto a mudanças e evoluções neste processo. Encarada o da forma aqui apresentada o enriquecimento da Base de Software pode evoluir em sua forma com manutenção da validade de seu conteúdo.

A metodologia e seu faseamento são mais importantes para a função gerencial e para Barantia de Qualidade. O ADS precisa deixar disponível para esta função tantas representações gráficas quantas sejam necessárias como, por exemplo, DFD, DER, DEM, DEI, RP, etc. Cada representação gráfica coloca a integração natural do conhecimento nele representado de modo que o DD continue a ser o agente definidor e integrador de todos os modelos.

#### 7. MANUTENÇÃO DE SOFTWARE

A função Manutenção de Sistemas admite vários escopos.

Na verdade poderíamos pensar que desde a menção das modificações até a adição de uma grande nova função em um sistema existente e tudo pode ser considerado como Manutenção de Sistemas.

Para ADS, com veículo de aquisição de conhecimento para construir uma sólida e permanente Base de Software, a escolha do escopo da função Manutenção não muda praticamente nada. Todavia, para seus aspectos de planejamento e controle, estabelecimento de prioridades e alocação de recursos, esta escolha é fundamental. No momento em que 80% do trabalho do ADS é manutenção e 20% novo desenvolvimento, a função gerencial fica bastante prejudicada. O que se sugere é que só seja considerado "manutenção" ações corretivas inadiáveis em

Sistemas implantados. Toda e qualquer outra atividade deve ser identificada com o um projeto e percorrer todos os pontos de controle do ADS modelando e especificando cuidadosamente as mudanças e objetos, após ser submetido ao processo de exame de sua prioridade, ela borçação de programa de trabalho e conveniente, adequada e realista alocação de recursos necessários.

De qualquer forma, o esforço da função manutenção é atualizar representações gráficas e especificação de objetos, enriquecendo e atualizando a Base de Software. Os instrumentos disponíveis para a função manutenção são os mesmos alocados às diversas fases do Desenvolvimento. O que o ADS tem que proporcionar, a mais para a manutenção, é controle sobre as alterações que promove.

#### CONCLUSÃO

O problema do desenvolvimento sistemático de software, ou seja, a engenharia de software, requer o uso de um ambiente de desenvolvimento de software. Este importa a cultura local, o processo usual de desenvolvimento, a gerência do desenvolvimento e a interação com o usuário. Requer, para alcançar sucesso o emprego de ferramentas, capacitação da mão-de-obra e uma gerência eficaz. Em resumo, o problema da baixa qualidade e produtividade é essencialmente um problema de gerência e do não emprego de ferramentas eficazes.

#### REFERÊNCIAS BIBLIOGRÁFICAS:

- Humphrey, W. S.: "Characterizing the Software Process: A Maturity Framework" Zeilkowitz, M. V.; Yen, R. T.; Hamlet, R. G.; Gannon, J. D.; Basilli, U. R.: "Software Engineering Practices in US and Japan" IEEE, June 1984
- Lucena, C. J. P.: Takahashi, E. T.: "A Anatomia de um Ambiente de Desenvolvimento de Software" PUC - RJ - Departamento de Informática - 1989
- Brooks, F. P.: "The Mythical Man Month" Addison Wesley Co., 1975
- DRRI, A. S.; Ellison, R. J.; Feiler, P. H.; Habermann, A. N.: "Software Development Environment" IEEE Computer, novembro 1987
- Ritto, A. C. A.: "Desenvolvimento Planejado de Sistemas" UFPE, tese de Mestrado em Informática