

Nesta seção você encontra artigos voltados para testes, processo, modelos, documentação, entre outros



## Dificuldades e Problemas Encontrados na Manutenção de Software

O desenvolvimento de qualquer software, exceto programas muito simples, é uma tarefa bastante complexa. Esse fato se tornou evidente com a “crise de software”, o que originou o conceito de engenharia de software como uma abordagem sistemática, disciplinada e quantificável para o desenvolvimento, manutenção e descarte de software.

A engenharia de software surgiu inicialmente mais como uma promessa do que uma realidade. De fato, muitos dos problemas ligados ao desenvolvimento e manutenção de software continuam sem solução, apesar de muitos conceitos terem evoluído.

Entender o que significa manutenção de software, e principalmente a abrangência do significado do termo, constitui passo fundamental para o estudo e aprofundamento de soluções para os problemas atrelados a essa tarefa.

Neste contexto, este artigo foca na discussão sobre as dificuldades encontradas

### *De que se trata o artigo?*

Este artigo foca na discussão sobre as dificuldades encontradas na realização das atividades de manutenção de software. Para isso, apresenta a realização de um estudo de observação que foi conduzido com o objetivo de verificar os problemas de manutenção de software existentes em uma organização empenhada no desenvolvimento de software comercial.

### *Para que serve?*

Este texto é indicado para aqueles profissionais que atuam com manutenção de software no seu dia-a-dia e que desejam entender melhor os atuais desafios em torno desse tema.

### *Em que situação o tema é útil?*

A manutenção de software é hoje um assunto presente em organizações que desenvolvem e mantêm software. Isso se deve à necessidade de sempre ajustar e melhorar o produto de software de acordo com as mais diversas necessidades. Diante desse fato, entender o significado e abrangência do termo manutenção de software pode auxiliar organizações e profissionais interessados no tema a melhor conduzir seus esforços quando precisam manter seus produtos.



**Mateus Maida Paduelli**

Bacharel e mestre em computação pela Universidade de São Paulo. Já atuou em diversas organizações com o tema manutenção de software e atualmente é servidor público federal.

na realização das atividades de manutenção de software. Para isso, apresenta a realização de um estudo de observação que foi conduzido com o objetivo de verificar os problemas de manutenção de software existentes em uma organização empenhada no desenvolvimento de software comercial. Além disso, apresenta uma comparação entre os problemas de manutenção de software atuais, identificados pelo estudo de caso, e aqueles registrados no passado, buscando inferir algum padrão de evolução nos mesmos.

Antes de iniciarmos a discussão sobre as dificuldades e problemas encontrados nas atividades de manutenção de software, iremos apresentar rapidamente algumas definições associadas à área de manutenção.

### Atividade de Manutenção de Software

Este tópico tem por objetivo destacar as características, tipos e desafios para a manutenção de software. Trata-se de um tópico importante para esclarecer alguns pontos pertinentes ao assunto que serão considerados ao longo deste artigo.

### Definições

A atividade de manutenção de software é caracterizada pela modificação de um produto de software já entregue ao cliente, para a correção de eventuais erros, melhora em seu desempenho, ou qualquer outro atributo, ou ainda para adaptação desse produto a um ambiente modificado.

Embora a definição trate genericamente qualquer produto de software, existem diferenças entre a manutenção de softwares com propósitos distintos.

Uma primeira classificação representa aqueles softwares construídos com base em uma especificação rígida e bem definida, cujos resultados esperados são bem conhecidos. Por exemplo, um software construído para realizar operações com matrizes (adição, multiplicação e inversão). Nesse tipo de software, uma vez que tenha sido construído considerando a correta implementação do método, dificilmente haverá a necessidade de manutenções.

Já em uma segunda classificação, são agrupados os softwares que constituem implementações de soluções aproximadas para problemas do mundo real, uma vez que soluções completas somente são conseguidas na teoria nesses casos. Como exemplo, pode-se citar um jogo de xadrez. Embora suas regras sejam bem definidas, não é possível construir um software que calcule a cada passo todos os possíveis movimentos de peças do tabuleiro, de forma a determinar o melhor movimento. Isso porque o número de movimentos possíveis é muito grande para ser calculado em um intervalo de tempo relativamente curto. A técnica utilizada para desenvolver esse tipo de solução baseia-se em descrever o problema de forma abstrata e então definir os requisitos de software a partir dessa abstração.

Percebe-se que esse tipo de sistema já abre espaço para diferentes interpretações por parte do desenvolvedor, o que tende a produzir software com maior necessidade de manutenção do que quando comparado aos da classificação anterior. Por considerar uma abstração para especificação de requisitos, a

necessidade de mudança pode aparecer caso a abstração mude, na medida em que um maior entendimento do problema seja alcançado.

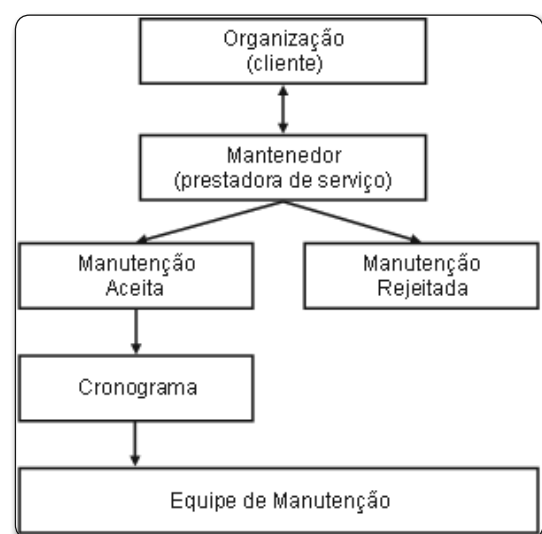
Finalmente, uma terceira e última classe de softwares considera mudanças no ambiente onde o software vai ser utilizado, característica não existente nas duas classificações anteriores.

Um software integrante da terceira classificação corresponde àquele criado com base em um modelo dos processos abstratos envolvidos no sistema, e precisará mudar sempre que ocorrer mudanças nesses modelos, sendo, portanto, parte do ambiente que ele modela. Um exemplo desse tipo de software seria aquele que apresenta informações da economia de um país. À medida que a economia passa a ser mais bem compreendida, o modelo muda e com ele a abstração do problema, causando uma necessidade inevitável de manutenção no software. Esse tipo de software, comumente encontrado no dia-a-dia das organizações, tem interesse particular neste trabalho.

Além de considerar tipos diferentes de software, o processo de manutenção não corresponde a uma atividade isolada. Durante a sua execução, diferentes partes precisam interagir de forma que os objetivos da manutenção sejam entendidos e os resultados esperados alcançados. Essas partes são:

- **Organização Cliente:** essa organização corresponde ao adquirente do software, conforme definido pela norma ISO/IEC 12207. Isso significa a organização que possui o software e requisita o serviço de manutenção.
- **Mantenedor:** trata-se da organização que oferece o serviço de manutenção.
- **Usuário:** representa a organização ou pessoa que utiliza o software em questão, valendo-se de suas funcionalidades para automatizar e facilitar tarefas.

Naturalmente, existe um relacionamento entre essas partes e a constituição de um fluxo para os pedidos de manutenção, conforme é esquematizado na **Figura 1**.



**Figura 1.** Fluxo envolvido na manutenção de software

A organização solicitante deve possuir algum responsável pela solicitação de manutenção, que será encarregado de verificar os requisitos e informar o mantenedor. Esse responsável deverá considerar os erros e dificuldades apontadas pelo help-desk, que corresponde ao departamento diretamente encarregado do diálogo com os usuários.

Do lado do mantenedor, a organização que presta o serviço deve possuir alguém responsável por avaliar as requisições, julgando-as apropriadas ou não para os objetivos do software e da organização solicitante. Uma vez que os pedidos de manutenção são aceitos, deve existir alguém responsável por estabelecer um cronograma de entrega, que deverá considerar as prioridades e interesses de ambas as partes. Esse cronograma deverá ser seguido pela equipe de manutenção, que compreende o pessoal envolvido diretamente em atender às solicitações.

Finalmente, o papel do usuário consiste em utilizar o software reportando problemas para o help-desk, que por sua vez informará o responsável pelas solicitações de manutenção, fechando o ciclo.

## Tipos de Manutenção

As ações ligadas à atividade de manutenção de software foram classificadas de acordo com sua natureza em três categorias: corretivas, adaptativas e perfectivas.

- **Manutenções do tipo corretivas visam corrigir defeitos de funcionalidade, o que inclui acertos emergenciais de programa.** Pfleeger (2001) expõe um exemplo desse tipo de manutenção, que consiste em um usuário apresentando um problema de impressão em um relatório. O número de linhas impresso por folha é muito grande, o que causa sobreposição de informações. O problema foi identificado como uma falha no driver da impressora, provocando a necessidade de se alterar o menu do relatório para aceitar um parâmetro adicional que determina o número máximo de linhas impressas por folha.

- **Manutenções do tipo adaptativas referem-se a adequar o software ao seu ambiente externo.** O exemplo apontado por Pfleeger (2001) ilustra bem essa categoria. Suponha um gerenciador de banco de dados, que faz parte um sistema maior de hardware e software. Em uma atualização do gerenciador, os programadores perceberam que as já existentes rotinas de acesso a disco precisavam agora de mais um parâmetro adicional. Essa manutenção corresponde a uma manutenção adaptativa, uma vez que teve por finalidade adequação do software ao seu ambiente e não a correção de um defeito.

- **Manutenções do tipo perfectivas têm por objetivo acrescentar novos recursos de funcionalidade ao software, normalmente em razão de solicitações dos usuários.** Significam ainda re-

projetar partes de um software, de forma a tornar mais simples a compreensão e utilização do mesmo. Como exemplo, pode-se citar o pedido do usuário por um novo relatório com informações que até então não podiam ser obtidas do banco de dados.

A união das categorias adaptativa e perfectiva é sugerida por Pigoski (1996), que propõe uni-las em uma única denominada aprimoramentos. Essa classificação estaria de acordo com organizações que geralmente utilizam o termo manutenção para se referir à execução de pequenas mudanças no software, enquanto o termo desenvolvimento é usado para os demais tipos de modificações.

Uma quarta categoria de manutenção é apresentada por alguns autores. Essa categoria se refere a manutenções do tipo preventivas que buscam identificar previamente possíveis fontes de problemas no software e corrigi-las antecipadamente.

A IEEE traz ainda uma categoria a mais chamada emergencial. Essa categoria é caracterizada pela execução de uma manutenção corretiva não planejada, com o intuito de manter o software operacional. Tal classificação insere a idéia de manutenção planejada e não-planejada, bem como manutenção reativa e pró-ativa, como é ilustrado na **Tabela 1**.

Para efeito de estudo, neste artigo será considerada a classificação de tipos de manutenção considerando: corretivas, adaptativas, perfectivas e preventivas.

## Estudo de Caso

A partir de agora apresentaremos um estudo de caso que foi conduzido com o objetivo de verificar os problemas de manutenção de software existentes em uma organização empenhada no desenvolvimento de software comercial. Essa organização mantém uma base de dados contendo registros históricos de manutenções efetuadas sobre um sistema de informação por ela desenvolvido e mantido.

## Descrição da Organização

Criada em 1993, a organização estudada corresponde a uma software-house, que desenvolve e mantém alguns softwares voltados à área médica e de saúde. É composta por uma unidade na cidade de São Carlos - SP, responsável pelo desenvolvimento e manutenção, e outra unidade na cidade de São Paulo, dedicada à venda e consultoria.

O software estudado corresponde a um sistema de porte médio voltado à odontologia, mais precisamente ao gerenciamento de operadoras odontológicas, sendo, no entanto, um produto com características típicas à grande maioria dos softwares que se sujeitam a gerenciar algum ramo de atividade comercial. São exemplos de recursos oferecidos pelo software: cadastro de clientes, cadastro de fornecedores, emissão de boletos bancários, controle financeiro, relatórios analíticos complexos, controle de estoques, suporte a multiusuário, controle de permissão de acesso a diferentes módulos, controle de fluxo de caixa, emissão de faturas, emissão de relatórios específicos para prestação de contas a órgãos de fiscalização do governo, integração com sistemas web etc.

	Planejada	Não-Planejada
Reativa	Corretiva Adaptativa	Emergencial
Pró-ativa	Perfectiva	

**Tabela 1.** Definições IEEE para as categorias de manutenção de software

Relativamente ao número de clientes, no momento da pesquisa, existiam 41 clientes (operadoras de odontologia), envolvendo desde organizações de pequeno porte, com até 5.000 associados cada, utilizando o software em cerca de 5 computadores, até empresas maiores, com mais de 50 máquinas em rede, utilizando o software de maneira simultânea e contando com até 150.000 associados. Normalmente essas empresas de maior porte são estruturadas por departamentos, e cada departamento utiliza um módulo específico do software.

## Dinâmica de Manutenção

Neste tópico são abordadas as características da maneira de trabalhar da organização, expondo como registra, altera e entrega as suas manutenções.

Uma observação inicial se faz necessária, e se refere à possível confusão entre o que seria uma tarefa de desenvolvimento e o que seria uma atividade de manutenção, no contexto do software estudado. De uma forma objetiva, o software aqui referido encontra-se desenvolvido, uma vez que está em uso por muitos clientes, ou seja, é um produto de software já entregue ao cliente. Partindo-se desse pressuposto, qualquer solicitação, seja por parte de clientes, seja por observação de algum profissional que atue sobre o software, será classificada como uma manutenção, ainda que exija criar um módulo novo ou refazer um já existente.

## Registro de Manutenções

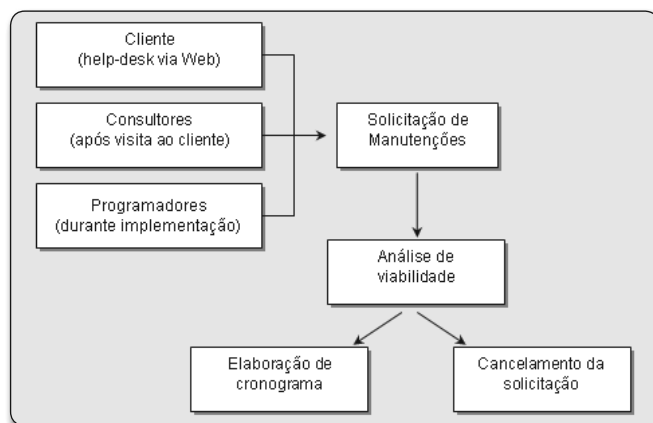
Os registros de manutenções são inseridos na base de dados de três formas distintas: (i) pelo próprio cliente, através do sistema de help-desk disponível no site da organização; (ii) pelos consultores da empresa, após visita ao cliente e levantamento de necessidades de manutenção; (iii) pelos próprios programadores, que podem identificar necessidades de manutenção à medida que “evoluem” o software. Uma interface específica baseada na web foi criada para a inserção desses registros na base.

Cada registro contém diversas informações, como por exemplo, qual o cliente que solicita, grau de prioridade, quem realizará a manutenção, tempo gasto previsto para a atividade, tipo de manutenção, status da implementação, datas de inserção do chamado e data prevista para entrega, observações, descrição da solução adotada, entre outras.

Após o registro de um chamado, sua execução dependerá da análise de viabilidade, efetuada pelo responsável pelo produto. O procedimento adotado está resumido na **Figura 2**.

Eventualmente, solicitações de manutenção são consideradas inviáveis, sendo canceladas pelo responsável por essa avaliação. Uma vez cancelada, esse fato é informado ao respectivo cliente, podendo ser revisada a necessidade para que uma proposta de manutenção mais adequada seja registrada.

Uma característica do sistema de controle de chamados é a interação que oferece entre mantenedor/cliente. Essa interação ocorre da seguinte maneira: quando o cliente registra alguma necessidade de manutenção, automaticamente a equipe de suporte e o analista responsável pelos cronogramas e testes,



**Figura 2.** Etapas envolvidas na solicitação de manutenções

Status do chamado	Cliente recebe e-mail
Pendente	Sim
Em execução	Sim
Cancelado	Sim
Em testes	Não
Em homologação	Sim
Concluído	Sim

**Tabela 2.** Situações possíveis para os chamados de manutenção

recebem um e-mail informando que uma nova solicitação foi feita. Diversos status são atribuídos a uma solicitação, e a cada troca de status, o cliente pode receber um e-mail informando o andamento de sua solicitação. Na **Tabela 2** estão representados os status possíveis para as solicitações, e em quais mudanças o cliente recebe um e-mail de notificação.

Na situação “em testes”, o cliente não recebe notificação, pois esse status é utilizado para controle interno. Uma solicitação “em testes” significa que está com a alteração no código-fonte efetuada e disponível para o analista responsável realizar os testes preliminares a fim de produzir a versão de homologação, que então é disponibilizada ao cliente. Somente quando a solicitação tiver sido testada e a versão de homologação estiver concluída e fornecida ao cliente, o analista responsável irá alterar o status para “em homologação”, permitindo então o envio da notificação ao cliente. A situação “concluído” ocorre quando a versão de homologação finalmente for instalada no ambiente de produção do cliente.

Para o status de “cancelado”, o cliente é informado apenas que sua solicitação foi negada, e que para maiores esclarecimentos ele deve entrar em contato com a equipe de suporte.

Por fim, destaca-se que o cliente pode, a qualquer momento, conferir os status de seus chamados, consultando sua área privativa no site da organização, que oferece acesso ao help-desk tanto para inclusão de novos chamados, como para acompanhamento por meio dos diferentes status.

## Controle de Versões

Os clientes têm à sua disposição uma área privativa no site da organização, através do qual podem realizar o

download de versões atualizadas do software, à medida que são disponibilizadas.

Essa disponibilização de versões ocorre mensalmente, podendo incluir rotinas de manutenção no banco de dados, quando necessárias, além dos componentes de software alterados. As atualizações de banco de dados são efetuadas automaticamente no momento em que o cliente abre o software pela primeira vez, após a atualização. Uma característica relevante é a de que o software sempre será o mesmo independente do cliente, ou seja, não existem versões distintas para clientes diferentes, todos utilizam uma versão única que pode ser configurada de acordo com necessidades específicas de cada cliente (módulo de opções de sistema do software).

Após a disponibilização de uma nova versão, o sistema de controle de chamados é consultado e as manutenções de software pendentes são consideradas, e então definidas quais entrarão na versão do mês seguinte. Essas manutenções são encaminhadas para os programadores (que consultam no sistema de chamados suas atividades pendentes), devendo seguir as datas programadas para entrega do módulo/arquivo alterado. As datas de entrega não são para o usuário final, mas sim para o analista responsável, que irá testá-las, e então disponibilizará uma versão de homologação do software. Essa versão é fornecida a um grupo de clientes com os quais foi estabelecido um acordo para realização desses testes. Isso ocorre na última semana do mês que antecede aquele de disponibilização da nova versão.

A idéia de disponibilizar versões intermediárias a um ou mais clientes para testes reais em ambiente à parte do de produção, corresponde a uma das técnicas de teste apresentadas por Pressman (2005), intitulada teste beta. Segundo o autor, esse é o tipo de teste conduzido nas instalações do próprio cliente, sem acompanhamento do desenvolvedor, estando o cliente responsável por registrar problemas e informá-los em intervalos regulares.

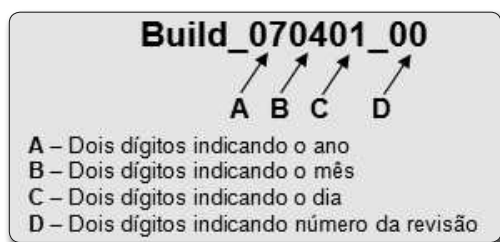


Figura 3. Padrão de numeração de versões

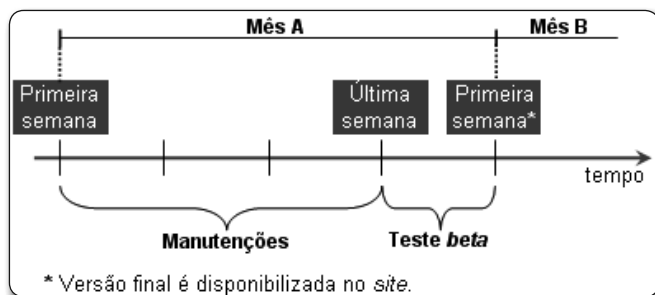


Figura 4. Linha do tempo para disponibilização de versões

Evidentemente entre a data de estabelecimento de cronograma para os chamados pendentes e a data de liberação da nova versão, surgem normalmente chamados novos e urgentes, que não podem esperar até próxima versão. No geral, procura-se evitar essa situação, mas quando ocorrem, versões intermediárias modificadas a partir da última versão oficial, somente com a alteração urgente ora solicitada, são disponibilizadas e apenas para o cliente que precisou da manutenção urgente.

Um padrão para nomenclatura das versões foi adotado, buscando atrelar a versão ao seu mês de lançamento, bem como informar o número da revisão atual da versão (caso a versão oficial tenha necessitado de alguma alteração emergencial). Na Figura 3 é ilustrado um exemplo do padrão de numeração adotado.

Os dois últimos dígitos são incrementados em uma unidade, à medida que a versão for passando por manutenções emergenciais (aquelas que não podem ser acumuladas para a versão do mês seguinte). No exemplo da Figura 3, a versão em questão é a oficial (revisão 00), do mês de abril de 2007.

A versão de homologação, uma vez testada pelo período de uma semana nos clientes selecionados, e com eventuais problemas detectados e corrigidos, é oficialmente convertida em uma versão final ("oficial"), e disponibilizada no site para todos os clientes, na primeira semana do mês seguinte ao de testes. Na Figura 4 é ilustrado esse processo.

Durante a fase de manutenção no código-fonte, é utilizado um software de controle de versão (Microsoft Source Safe), que trabalha da seguinte forma: cada desenvolvedor que for alterar algum arquivo (que faz parte do projeto do software) realiza uma operação de check-out desse arquivo do repositório (o qual armazena o projeto todo), ficando nesse momento "responsável" pelo arquivo, impedindo que qualquer outra pessoa possa realizar check-out do mesmo arquivo. Diz-se que o arquivo fica "travado" para o programador x. Somente quando a manutenção for concluída, o arquivo é submetido a um check-in, voltando para o repositório e ficando livre para uso por outro programador.

À medida que alterações vão sendo entregues e disponibilizadas no repositório, o analista responsável, realiza o check-out dos arquivos alterados e procede com testes preliminares. Uma vez que a data de liberação da nova versão para homologação se aproxima, esse analista realiza um *check-out* completo de todo o projeto do software, e realiza testes gerais, a fim de compor a versão de homologação para envio aos clientes pré-estabelecidos.

Durante a homologação, eventualmente esses clientes são acompanhados pessoalmente por algum consultor, ou remotamente via apoio por telefone, e-mail, VNC (software que permite conexão ao desktop remoto do cliente), entre outros.

Na medida em que problemas são verificados, se forem erros nas manutenções realizadas para a versão que será disponibilizada, esses erros são imediatamente corrigidos e a versão de homologação é atualizada. Se forem problemas novos (novas necessidades de alterações ainda não previstas), eles são registrados e então programados para disponibilização em versões posteriores do software.

## Metodologia

A metodologia utilizada para a execução do estudo de caso buscou, por um lado, estudar a base de dados na qual estão os registros de manutenção, e, por outro lado, entrevistar os profissionais responsáveis pela tarefa de manutenção. A forma como cada etapa foi conduzida está descrita a seguir.

## Parte A – Base de Dados

Na primeira etapa, o foco dos esforços centrou-se em extrair da base de dados totais a respeito de diferentes características das manutenções efetuadas no software. Esses totais foram obtidos por meio de consultas SQL à base, sendo que as totalizações buscadas estão resumidas na **Tabela 3**.

Finalmente, os números obtidos foram utilizados para construção de gráficos mostrados mais adiante neste artigo.

## Parte B – Questionário e Entrevista

A segunda etapa do levantamento de dados foi realizada com o auxílio de questionário e entrevistas.

O questionário elaborado visou obter dos profissionais envolvidos com manutenção de software características de seu trabalho diário, coletando uma gama de informações para análise posterior junto com os dados da Parte A.

Após a elaboração do questionário foram realizadas entrevistas individuais com essas pessoas, procurando-se registrar todo comentário e informação relevante para o propósito da entrevista.

## Análise dos Dados Coletados

A partir de agora apresentaremos os resultados das duas formas utilizadas para obtenção de dados: a pesquisa na base de dados e as entrevistas. A base de dados, no momento da pesquisa, contava com mais de 3700 solicitações de manutenção, e as entrevistas foram realizadas com 8 pessoas ligadas diretamente à manutenção do software.

## Estatísticas sobre a Base de Dados

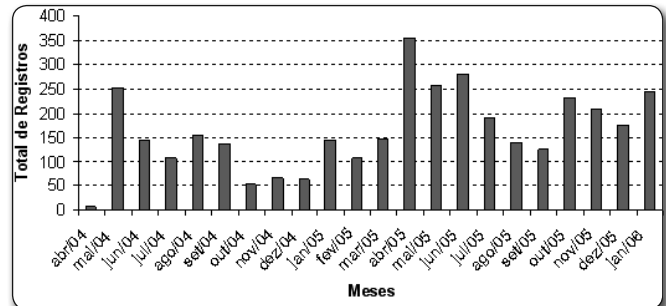
Partindo dos totais obtidos com a base de dados, algumas figuras puderam ser construídas.

Inicialmente, na **Figura 5**, é apresentado o número de solicitações registradas mensalmente desde a implantação do sistema de registro de manutenções.

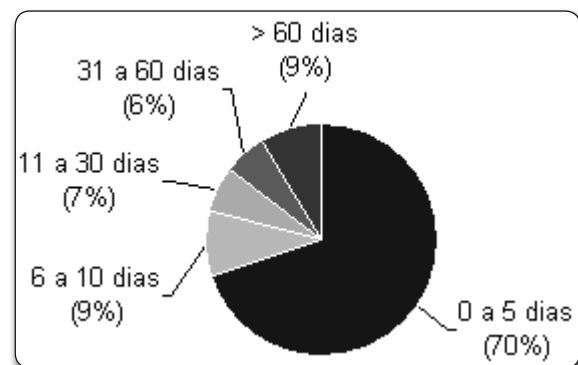
Nota-se que não existe uma constância no número de manutenções por mês, o que sugere uma imprevisibilidade nas necessidades de manutenção por parte dos clientes. Em abril de 2005 foi registrado o maior número de chamados de manutenção (355). Os dados representam uma média de 163,45 solicitações de manutenção por mês (desvio padrão de 83,7). Constatou-se ainda que a média de solicitações diárias é de 8,59 novas requisições. Uma possível razão para a existência de épocas com maior incidência de pedidos de manutenção seria o fato de o número de solicitações entregues naquele mês ter sido maior, o que normalmente acarretaria mais falhas colaterais, ou seja, aquelas que acabam incidindo em outros módulos do software, apesar dos testes e da homologação em clientes.

Valor	Objetivo
Distribuição de solicitações mensalmente	Verificar algum padrão na distribuição
Tempo em dias para entrega de solicitações	Verificar o tempo de resposta ao cliente
Totais de solicitações por grupo de prioridade	Verificar a expectativa do usuário
Totais por tipo de manutenção	Verificar distribuição entre os tipos
Tempo de desenvolvimento e manutenção	Verificar esforço necessário em cada fase

**Tabela 3.** Características de manutenção consideradas e seus objetivos



**Figura 5.** Total mensal de solicitações de manutenção



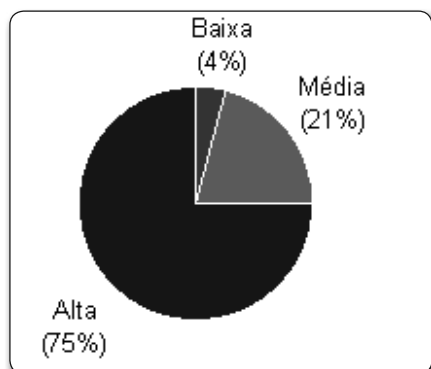
**Figura 6.** Dias necessários para a conclusão de manutenções

Os registros de solicitações de manutenção incluem desde operações simples, que resultam em poucos dias para a implementação, até solicitações mais complexas, que podem levar semanas, e até meses, para serem concluídas. Solicitações desse porte normalmente exigem que o projeto de um ou mais módulos seja reestruturado, o que justifica o tempo gasto para entrega ao cliente.

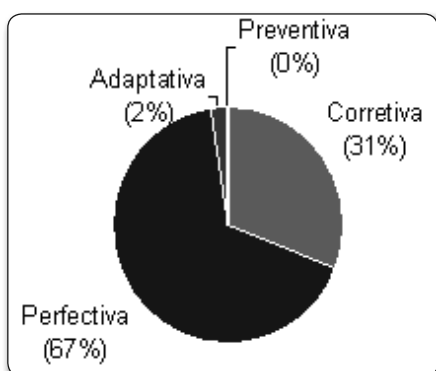
A distribuição, em termos de dias gastos para a implementação de manutenções, é mostrada na **Figura 6**.

Um estudo do gráfico revela que as manutenções de menor grau de complexidade ocupam a maior parte do tempo dos mantenedores. No entanto, existem manutenções com complexidade suficiente para exigir mais de dois meses de trabalho.

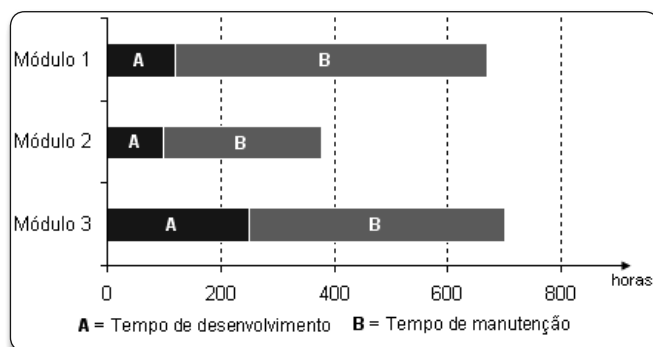
Observou-se que a grande quantidade de manutenções de menor complexidade acaba por interferir no tempo de entrega das manutenções mais complexas, gerando atrasos. Isso ocorre em função da grande expectativa do cliente no sentido de obter respostas rápidas, o que força os mantenedores a liberar primeiro as manutenções mais simples. Na **Figura 7** é mostrado qual o grau de prioridade indicado pelo cliente ao fazer as suas solicitações de manutenção.



**Figura 7.** Distribuição de prioridades das manutenções



**Figura 8.** Distribuição dos problemas entre os tipos de manutenção



**Figura 9.** Tempo consumido em desenvolvimento versus em manutenção

Nota-se que o cliente busca respostas rápidas, considerando com prioridade elevada a maior parte dos chamados de manutenção efetuados.

Do ponto de vista dos quatro tipos de manutenção software (adaptativa, perfectiva, corretiva e preventiva), realizou-se a quantificação das manutenções efetuadas dentro desses tipos. Para esse levantamento, foi considerada a classificação das manutenções informada em cada registro feito na base de dados. O resultado obtido é apresentado na **Figura 8**.

Os valores apresentados apontam que o maior esforço da organização está em atender requisições de novas funcionalidades, mostrando enfoque em manutenção perfectiva. Uma constatação preocupante foi a de que não existe uma atenção

em avaliar e prevenir problemas futuros no software (0% em manutenção preventiva), o que poderia ser feito por um processo de reengenharia. Fica evidente que o enfoque está em manter o software em funcionamento, adequando-o na medida em que as necessidades surgem, ou seja, seria uma postura de “aguardar acontecer” e não de “buscar prevenir”.

É esperado, pela literatura, que o tempo empenhado em tarefas de manutenção exceda o tempo de desenvolvimento, e esse fato buscou ser comprovado pela comparação entre tempo gasto com o desenvolvimento da primeira versão entregue ao cliente e o tempo gasto com a manutenção até então. Para isso, elegeam-se três módulos representativos do software analisado (aqui referenciados por módulo 1, módulo 2 e módulo 3), com o intuito de contabilizar o tempo gasto em desenvolvimento e o gasto em manutenção. O tempo de desenvolvimento inclui tempo gasto em projeto, codificação e testes.

Esse resultado é apresentado na **Figura 9**, que considera os registros de manutenção efetuados.

De acordo com o mostrado na figura, percebe-se que a organização consumiu, nos três exemplos, mais de 60% do esforço empregado em atividades de manutenção, e esse percentual tende a subir com o passar do tempo.

### Ambiente e Equipe de Manutenção

Durante a entrevista com os responsáveis pelas manutenções, constatou-se que não existe um procedimento padrão a ser seguido para a execução das manutenções. O que existe é uma preocupação em agendar datas de entrega para as solicitações cujos clientes responsáveis insistem em respostas mais rápidas, o que frequentemente acaba inviabilizando a metodologia de disponibilização de atualizações descrita anteriormente. Casos de manutenções de urgência, quando ocorrem, são priorizados em detrimento das demais.

Observou-se também que o código-fonte modificado não é documentado de maneira adequada, sendo muitas vezes atribuídas somente pequenas notas, ou o número do chamado que resultou na modificação de um trecho de código. Esse número de chamado está relacionado com o registro de necessidade de manutenção efetuado pelo cliente, podendo um futuro mantenedor consultar do que se tratou a manutenção, buscando pelo número do chamado informado no código-fonte no sistema de registro de manutenções.

Os profissionais encarregados de tarefas de manutenção relataram ainda que nem sempre o cliente compreende totalmente o que está solicitando, muitas vezes gerando discussões entre empresa/cliente, o que está relacionado com os problemas de comunicação entre usuário e desenvolvedor. Muitas vezes o cliente acredita que uma determinada manutenção será suficiente para adequar o software à sua nova necessidade de negócio, quando na verdade não é suficiente. Essa dificuldade de compreensão do software pelo próprio cliente acaba por gerar situações de desgaste na relação entre empresa-cliente.

Prazo de entrega é sempre um problema quando não existe um processo para conduzir a manutenção. Esse foi outro fato verificado, uma vez que nem sempre a manutenção, com data

agendada e informada ao cliente, pode ser entregue no prazo combinado, geralmente em função de re-trabalho em manutenções já realizadas, ou mudanças de prioridades de entrega.

Em parte esse atraso decorre do fato de o mantenedor ser também a pessoa que desenvolve, de forma que a tarefa de desenvolvimento precisa mesclar-se com a de manutenção, contribuindo para a diminuição do desempenho do mantenedor. Isso se agrava quando testes em uma nova funcionalidade estão sendo feitos em paralelo a alguma correção solicitada pelo cliente.

Outro ponto importante refere-se ao baixo interesse dos profissionais envolvidos em tarefas de manutenção, fato que pode ser comprovado pelo interesse desses profissionais em deixar a atividade de manutenção para dedicar-se ao desenvolvimento. Esse interesse foi manifestado por todos os profissionais de manutenção entrevistados.

A política de testes utilizada, embora prevista como uma alternativa pela literatura, nesse caso específico não vinha apresentando os resultados esperados. Nem sempre o cliente tinha tempo e boa vontade para avaliar uma versão de testes em ambiente separado daquele de produção, culminando com problemas nas manutenções efetuadas surgindo quando a versão oficial já estava no site e em uso pela maior parte dos clientes em seus ambientes de produção.

## Problemas Identificados

Após obter e analisar os dados, foi possível montar a **Tabela 4**, que envolve os principais problemas de manutenção identificados na organização.

Os problemas, como pode ser observado, foram classificados de acordo com sua natureza em problemas gerenciais e técnicos.

## Conclusões Parciais

A classificação dos problemas de manutenção identificados na organização quanto a sua natureza, permite observar que aqueles ligados a questões gerenciais estão mais presentes do que os de caráter mais técnico. De fato, constatou-se que conciliar dificuldades técnicas de manutenção com os anseios dos usuários não é uma tarefa simples, e infelizmente não vem obtendo o sucesso desejado.

A ausência de um processo de manutenção de software formal, como dita normas de engenharia de software, não seria de todo condenável se o processo interno criado pela própria organização funcionasse de maneira satisfatória. O que se observou foi que a pressão gerada pelos clientes muitas vezes prejudica o cumprimento correto da seqüência de passos estipulada pela própria organização para tratar a manutenção de software.

Por fim, a inexistência de manutenção preventiva é um fato que precisa ser analisado em conjunto com os objetivos de negócio da organização e com seus projetos em relação à vida útil do software que mantém.

## Problemas de manutenção de software

Este capítulo apresenta uma comparação entre os problemas de manutenção de software atuais, identificados pelo estudo

de caso, e aqueles registrados no passado, buscando inferir algum padrão de evolução nos mesmos. Além disso, estabelece um rol de problemas de manutenção de software, publicados por diferentes autores, na forma de uma lista de dificuldades inerentes à atividade de manutenção, para fins de análise nos próximos capítulos. Procurou-se estabelecer uma denominação direta e simples para os problemas, facilitando a leitura, o entendimento e posteriores referências.

## Problemas do Passado versus Problemas Atuais

Os problemas identificados no estudo de caso permitem o surgimento da seguinte questão: Os problemas de manutenção de software de hoje guardam relação com aqueles verificados no passado? A resposta a essa questão fornecerá indícios da maneira como a mudança da tecnologia pode estar ou não influenciando tanto o agravamento como o surgimento de problemas.

Partindo desse princípio, constatou-se que existe um alinhamento mais ou menos regular entre os problemas do passado com os atuais, o que permitiu a construção da **Tabela 5**. O quadro associa os problemas relacionando-os de acordo com a sua natureza.

Embora não se estabeleça uma equivalência exata, essa tabela mostra uma semelhança muito grande entre a natureza dos problemas obtidos em cada levantamento.

## Problemas de Manutenção Publicados

A relação de problemas de manutenção de software identificada pode ser visualizada na **Tabela 6**, construída de forma a mostrar os problemas e suas respectivas fontes. Em virtude

Problemas Gerenciais	Ausência de um processo de manutenção de software
	Grande expectativa dos usuários
	Elevada rotatividade de membros e funções dentro da equipe
	Sobrecarga de tarefas
	Estimativa de prazo não condizente com a complexidade do software
	Baixa motivação entre profissionais de manutenção
	Ausência de manutenção preventiva
	Falhas de comunicação com o usuário
Problemas Técnicos	Atrasos na entrega
	Registro inexistente ou superficial de manutenções anteriores
	Ausência de um ambiente computacional específico para manutenção
	Validação insuficiente de manutenções efetuadas
	Documentação insuficiente ou superficial
	Falta de compreensão do software e suas estruturas

**Tabela 4.** Problemas de manutenção de software



da lista relativamente extensa, optou-se por dividi-la em categorias para promover melhor organização e facilidade de leitura. A numeração sequencial atribuída a cada problema tem o intuito único de facilitar as referências posteriores, não guardando qualquer outro significado. Faz-se necessário dizer que a separação dos problemas a seguir não é rigorosamente exata, uma vez que muitos deles poderiam ser classificados em mais de uma categoria.

A relação anterior, embora seguramente não exaustiva, apresenta de forma geral os principais problemas que recaem sobre a atividade de manutenção de software. Outros problemas que eventualmente existam e não foram citados, são demasiados específicos de certo domínio, ou fortemente relacionados a algum dos problemas já citados. Dessa forma, acredita-se que a listagem anterior seja representativa das dificuldades mais importantes em manutenção de software, não se prendendo a nenhum domínio específico de software.

## Problemas de manutenção e os grupos de processos da norma ISO/IEC 12207

A norma ISO/IEC 12207 representa um padrão internacionalmente adotado que engloba processos para o ciclo de vida de software. Neste tópico, a norma é utilizada para criar uma relação de causa e consequência com os problemas de manutenção de software apresentados anteriormente.

Essa relação se estabelece da seguinte maneira: a norma apresenta grupos de processos (que por sua vez englobam diversos outros processos), que visam atingir alguma fase do ciclo de vida de software, guiando-o com as melhores práticas. Assim, a não verificação de um grupo de processos, acarretará problemas para a respectiva fase (e consequências para as futuras).

Dessa maneira, foi possível a construção da **Tabela 7**, na qual são relacionados os grupos de processos da norma, com os problemas de manutenção, procurando estabelecer uma relação de causa (a não verificação do respectivo grupo) e consequência (problema derivado). Para essa distribuição, verificaram-se,

para cada grupo, seus processos e as determinações de suas respectivas tarefas. O não atendimento a uma das tarefas do processo, automaticamente relaciona o problema ao processo, e em um nível mais alto, ao grupo ao qual esse processo pertence.

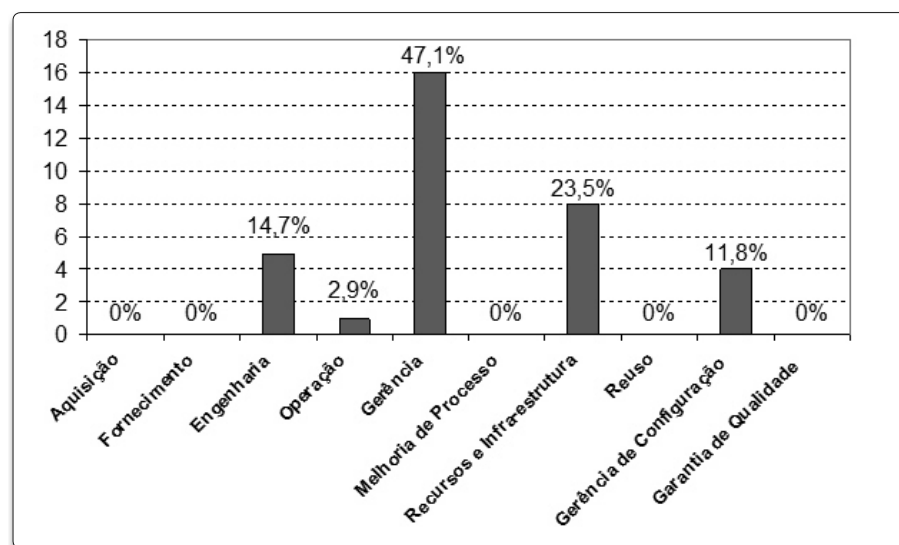
A distribuição gráfica das informações anteriores está representada na **Figura 10**.

Percebe-se, pela figura anterior, que existe uma grande concentração de problemas relacionados principalmente com fatores de ordem gerencial, ligados à interação com pessoas e processos, respondendo por 47,1% dos problemas.

Dentre os dez grupos de processos, cinco ficaram sem nenhum problema relacionado. Os grupos de aquisição e de fornecimento apresentam uma razão clara: estão relacionados à criação de um produto de software novo, portanto fortemente ligadas ao desenvolvimento, o que não foi considerado. O grupo de melhoria de processo pode ser visto sob duas perspectivas, que consideram a existência ou não de processos implantados na organização. Na primeira das visões, assumindo que a organização possua processos implantados, os problemas de manutenção poderiam ser associados a falhas nesses processos, alegando-se que seriam essas falhas as responsáveis diretamente pela maioria dos problemas de manutenção. Isso, no

Problemas apontados por Lientz e Swanson (1980)	Problemas atuais
Baixa qualidade da documentação dos sistemas	- Registro inexistente ou superficial de manutenções anteriores; - Documentação insuficiente ou superficial.
Necessidade constante dos usuários por melhorias e novas funcionalidades	- Grande expectativa dos usuários; - Falhas de comunicação com o usuário.
Falta de uma equipe de manutenção	- Ausência de um processo de manutenção de software; - Sobrecarga de tarefas; - Ausência de manutenção preventiva; - Ausência de um ambiente computacional específico para manutenção.
Falta de comprometimentos com cronogramas	- Estimativa de prazo não condizente com a complexidade do software; - Atrasos na entrega.
Treinamento inadequado do pessoal de manutenção	- Baixa motivação entre profissionais de manutenção; - Validação insuficiente de manutenções efetuadas; - Falta de compreensão do software e suas estruturas.
Rotatividade dos profissionais	- Elevada rotatividade de membros e funções dentro da equipe.

**Tabela 5.** Comparação entre problemas de manutenção de software



**Figura 10.** Distribuição dos problemas nos grupos de processos

Fatores de Gerência	
Problema	Referências
01. Falta de comprometimento com cronogramas	Lientz e Swanson (1980); Estudo de Caso
02. Treinamento inadequado da equipe de manutenção	Lientz e Swanson (1980); Dekleva (1992); Dart et al. (2001); Estudo de Caso
03. Ausência de um profissional responsável exclusivamente ao controle de configuração de software	Dart et al. (2001)
04. Visão organizacional diferenciada para a equipe de desenvolvimento e para a equipe de manutenção	Dart et al. (2001)
05. Dificuldades de comunicação entre a equipe de manutenção e a própria organização	Dart et al. (2001)
06. Software desenvolvido é visto pela gerência como não construído para a manutenção	Dart et al. (2001)
07. Mantenedores desconhecem planos da organização com relação à equipe de manutenção	Dart et al. (2001)
08. Contratação de temporários para auxílio na manutenção leva ao menor controle sobre a atividade	Dart et al. (2001)
09. Experiências com manutenções anteriores não são disseminadas dentro da própria organização e entre novos membros da equipe	Dart et al. (2001); Estudo de Caso
10. Dificuldade na medição do desempenho da equipe de manutenção	Dekleva (1992)
11. Ausência de adoção de padrões, metodologias e procedimentos de manutenção	Dekleva (1992); Estudo de Caso
12. Falta de suporte da gerência	Dekleva (1992)
13. Sobrecarga de tarefas	Estudo de Caso
14. Ausência de manutenção preventiva	Estudo de Caso
15. Estimativa de prazo não condizente com a complexidade do software	Estudo de Caso
Fatores de Infraestrutura	
Problema	Referências
16. Necessidade de suporte automatizado à gerência de configuração de software	Dart et al. (2001)
17. Ferramentas para manutenção precisam ser diferentes daquelas de desenvolvimento	Dart et al. (2001)
18. Falta de recursos tecnológicos adequados	Dart et al. (2001); Estudo de Caso
19. Estagnação tecnológica no ambiente de trabalho	Dart et al. (2001)
Fatores humanos – equipe	
Problema	Referências
20. Falta de uma equipe de manutenção	Lientz e Swanson (1980)
21. Elevada rotatividade dos profissionais	Lientz e Swanson (1980); Dekleva (1992); Estudo de Caso
22. Mantenedores lamentam por não possuírem contato com estado-da-arte da tecnologia	Dart et al. (2001)
23. Preferência dos profissionais por trabalhos de desenvolvimento	Dart et al. (2001)
24. Manutenção de software é vista como uma tarefa não prestigiosa	Dekleva (1992); Dart et al. (2001); Estudo de Caso
25. Falhas de comunicação com o usuário	Estudo de Caso
26. Métodos inadequados de testes	Dekleva (1992); Estudo de Caso
27. Documentação insuficiente ou superficial (software alterado)	Lientz e Swanson (1980); Dekleva (1992); Dart et al. (2001); Estudo de Caso
Fatores humanos – cliente	
Problema	Referências
28. Necessidade constante dos usuários por melhorias e novas funcionalidades	Lientz e Swanson (1980); Estudo de Caso
29. Falta de compreensão dos usuários a respeito de suas reais necessidades de software	Estudo de Caso
30. Mudanças freqüentes de prioridades (clientes)	Dekleva (1992)
Fatores de software	
Problema	Referências
31. Baixa qualidade da documentação dos sistemas (software original)	Lientz e Swanson (1980); Dekleva (1992); Dart et al. (2001); Estudo de Caso
32. Má qualidade do código-fonte original	Dekleva (1992); Dart et al. (2001)
33. Necessidades de integração com softwares incompatíveis	Dart et al. (2001)
34. Plataformas heterogêneas dificultam a definição de ferramentas adequadas	Dart et al. (2001)

Tabela 6. Problemas de manutenção de software – lista geral

	Problemas
<b>Categoria Processos Fundamentais</b>	
Grupo de Processos de Aquisição	
Grupo de Processos de Fornecimento	
Grupo de Processos de Engenharia	(11), (26), (29), (33), (34)
Grupo de Processos de Operação	(25)
<b>Categoria Processos Organizacionais</b>	
Grupo de Processos de Gerência	(01), (04), (05), (06), (07), (08), (12), (13), (14), (15), (20), (21), (24), (30), (31), (32)
Grupo de Processos de Melhoria de Processo	
Grupo de Processos de Recursos e Infra-estrutura	(02), (09), (10), (17), (18), (19), (22), (23)
Grupo de Processos de Reuso	
<b>Categoria Processos de Apoio</b>	
Grupo de Processos de Gerência de Configuração	(03), (16), (27), (28)
Grupo de Processos de Garantia de Qualidade	

**Tabela 7.** Grupos de processos não observados e as respectivas consequências

entanto, não corresponde à realidade, já que normalmente os processos não existem nas organizações.

Em outra visão do grupo de melhoria de processo, agora considerando que a organização não possua processos implantados, seria inadequado valer-se desse fato para se associar a esse grupo praticamente todos os problemas de manutenção, sob a alegação de que existem como resultado da ausência de processos adequados para tratar a manutenção de software. A inadequação desse posicionamento se justifica pelo fato de que é possível estabelecer uma melhor relação dos problemas com outros processos da norma, muitas vezes de forma direta, valendo-se para isso das tarefas estipuladas para cada processo. O não cumprimento de uma tarefa específica, automaticamente relaciona o processo correspondente com algum dos problemas já apresentados. Assim, pode-se concluir que a norma é capaz de fornecer indícios de quais processos específicos inexistem ou são falhos para a ocorrência de cada problema, permitindo então distribuí-los de forma mais precisa e não associá-los todos ao grupo de melhoria de processo sob a alegação genérica de sua ausência na organização.

O próximo grupo a considerar é o de reuso, que também parece ter uma razão clara para ter permanecido sem problemas associados. Não é simples tratar o reuso de artefatos de software nem mesmo durante o desenvolvimento. O que dizer então da atividade de manutenção, normalmente com necessidades imprevisíveis e muitas vezes inéditas para o contexto do software. Difícilmente será possível valer-se de soluções prontas, o que torna o grupo de processos de reuso com pouca ou nenhuma importância para a redução de problemas de manutenção de software.

Finalmente, o grupo de processos de garantia de qualidade também não teve problemas relacionados em razão

de existirem outros processos da norma com tarefas específicas não atendidas que melhor se encaixam como fonte do problema. Por exemplo, considere o processo de garantia de qualidade (os processos do grupo são: garantia de qualidade, verificação, validação, revisão conjunta, auditoria e avaliação de produto). O objetivo principal desse processo é prover garantias de que os produtos de trabalho e processos obedeçam aos planos predefinidos para o software. Note, no entanto, que os problemas de manutenção de software normalmente são frutos da ausência de planejamento ou do planejamento falho da manutenção, sendo que o correto planejamento e execução são tarefas reservadas a outro processo (processo de Manutenção de Software e Sistema). Assim, a existência de processos com tarefas mais específicas acabou por evitar que problemas de manutenção fossem relacionados a processos do grupo de garantia de qualidade.

No entanto, é fácil perceber que a palavra qualidade é extremamente conhecida em diversos contextos, mesmo aqueles não relacionados a software, chegando a ser quase intuitivo pensar que algo feito com qualidade resultará em menos problemas ou maior satisfação do usuário. Esse raciocínio leva à conclusão de que, embora sem problemas associados diretamente, os processos do grupo de garantia de qualidade devem ser observados como recurso de apoio à execução com qualidade das tarefas de manutenção como um todo.

## Considerações Finais

Embora tenha se produzido neste artigo uma relação de problemas com base em apontamentos feitos por outros autores, e também pelo estudo de caso realizado, entende-se que não se trata de uma relação fixa e exaustiva. Em diferentes abordagens futuras, nomes diferentes podem ser utilizados para se referirem às mesmas dificuldades. É razoável pensar ainda que novos problemas surjam, constituindo novas categorias de problemas, distintas das aqui definidas.

Esse fato, porém, não invalida ou torna de todo desatualizado o que foi exposto, já que a literatura, e também o estudo de caso apresentado, apontam uma relativa estabilidade entre os problemas mais relevantes ao longo do tempo.

É admissível, também, que o surgimento de novas técnicas de programação, novas tecnologias e metodologias, influenciem a forma como as organizações trabalham, e nesse caso poderiam reduzir os problemas aqui apresentados.

Outra observação necessária se faz com relação aos problemas e os tamanhos de software sobre os quais incidem. O que ocorre, de fato, é que os problemas apresentados incidem de maneira geral sobre qualquer tamanho de software, variando apenas a ênfase que o problema representa para cada tamanho de software.

Por fim, a relação estabelecida entre os problemas e os grupos da norma, permite uma visão inicial das características do processo de ciclo de vida de software que devem ser tomadas com maior atenção para o contexto de minimização de problemas de manutenção de software.

## Dê seu feedback sobre esta edição!

A Engenharia de Software Magazine tem que ser feita ao seu gosto. Para isso, precisamos saber o que você, leitor, acha da revista! Dê seu voto sobre este artigo, através do link:

[www.devmedia.com.br/esmag/feedback](http://www.devmedia.com.br/esmag/feedback)



## Referências

- Bennett, K. H.; Rajlich, V. T. (2000) "Software maintenance and evolution: a roadmap", In: Conference on The Future of Software Engineering, Limerick, Ireland, June.
- Bennett, K. H.; Ramage, M.; Munro, M. (1999) "Decision Model for Legacy Systems", IEEE Proceedings on Software (TSE), v. 146, n. 3, p. 153-159.
- Bhatt, P.; Shroff, G.; Misra, A. K. (2004) "Dynamics of software maintenance", ACM SIGSOFT Software Engineering Notes, v. 29, n. 5, p. 1-5.
- Basili, V. (1990) "Viewing Maintenance as Reuse-Oriented Software Development", IEEE Software, v. 7, n. 1, p. 19-25.
- Chapin, N. (1986) "Software maintenance: A different view", In: Conference on Software Maintenance, Orlando, FL, USA, November.
- Dart, S.; Christie, A. M.; Brown, A. W. (2001) "A Case Study in Software Maintenance", Technical Report, Carnegie Mellon University.
- Dekleva, S. M. (1990) "Annual Software Maintenance Survey: Survey Results", Software Maintenance Association, Vallejo, California, USA.
- \_\_\_\_\_. (1992) "Delphi study of software maintenance problems", In: Conference on Software Maintenance, Orlando, FL, USA, November.
- De Lucia, A.; Pompella, E.; Stefanucci, S. (2004) "Effort estimation for corrective software maintenance", In: 14th International Conference on Software Engineering and Knowledge Engineering, Ischia, Italy, July.
- IEEE (1998) "Std 1219 – IEEE Standard for Software Maintenance", Institute of Electrical and Electronic Engineers, New York, NY, USA.
- ISO/IEC 12207 (1998) "Standard for Information Technology - Software Lifecycle Processes", International Standard Organization, New York, NY, USA.
- ISO/IEC 15504 (2003) "Software Process Assessment", International Standard Organization, New York, NY, USA.
- Koskinen, J.; Salminen, A.; Paakki, J. (2004) "Hypertext support for the information needs of software maintainers", Journal of Software Maintenance and Evolution: Research and Practice, v. 16, n. 3, p. 187-215.
- Kung, H.; Hsu, C. (1998) "Software Maintenance Life Cycle Model", In: International Conference on Software Maintenance, Bethesda, Maryland, USA.
- Lehman, M. M. (1974) "Programs, Cities, Students, Limits to Growth?", In: Imperial College of Science Technology, London, England, May.
- \_\_\_\_\_. (1980) "On Understanding Laws, Evolution and Conservation in the Large Program Life Cycle", Journal of Systems and Software (JSS), v. 1, n. 3, p. 213-221.
- \_\_\_\_\_. (1991) "Software Engineering, the Software Process and their Support", IEEE Software Engineering Journal: Special Issues on Software Environments and Factories, v. 1, n. 3, p. 213-221.
- \_\_\_\_\_. (1996) "Laws of Software Evolution Revisited", In: 5th European Workshop on Software Process Technology, Nancy, France, October.
- Lientz, B. P.; Swanson, E. B. (1980) "Software Maintenance Management", Reading, MA, Addison-Wesley.
- Niessink, F. (1999) "Software Maintenance Research in the Mire?", In: Annual Workshop on Empirical Studies of Software Maintenance (WESS'99), Oxford, United Kingdom, September.
- Pfleeger, S. L. (2001) "Software Engineering: theory and practice", Second Edition, New Jersey, Prentice Hall.
- Pigoski, T. M. (1996) "Practical Software Maintenance: Best Practices for Managing Your Software Investment", Wiley Computer Publishing.
- Polo, M.; Piattini, M.; Ruiz, F.; Calero, C. (1999) "Roles in the maintenance process", ACM SIGSOFT Software Engineering Notes, v. 24, n. 4, p. 84-86.
- Polo, M.; Piattini, M.; Ruiz, F. (2003) "Using a qualitative research method for building a software maintenance methodology", Software – Practice and Experience, v. 32, n. 13, p. 1239-1260.
- Pressman, R. S. (2005) "Software Engineering: a practitioner's approach", 6.ed., McGrawHill Higher Education.
- Singh, R. (1996). "International Standard ISO/IEC 12207 Software Life Cycle Processes", Software Process Improvement and Practice, vol. 2, p. 35-50.
- Sneed, H. M. (2003) "Critical Success Factors in Software Maintenance", In: International Conference on Software Maintenance, Amsterdam, The Netherlands, September.
- Sommerville, I. (2003) "Engenharia de Software", 6.ed., São Paulo, Addison Wesley.
- Silva, L. de P.; Santander, V. F. A. (2004) "Uma Análise Crítica dos Desafios para Engenharia de Requisitos em Manutenção de Software", In: Workshop em Engenharia de Requisitos, Tandil, Argentina, Dezembro.
- Souza, S. C. B. de; Neves, W. C. G. das; Anquetil, N.; Oliveira, K. M. de. (2004) "Documentação Essencial para Manutenção de Software II", In: I Workshop de Manutenção de Software Moderna, Brasília, DF, Brasil, Outubro.
- Ulrich, W. M. (1990) "The evolutionary growth of software reengineering and the decade ahead", American Programmer, v. 3, n. 10, p. 14-20.
- Visaggio, G. (2001) "Assessing the Maintenance Process Through Replicated, Controlled Experiment", Journal of Systems and Software, v. 44, n. 3, p. 187-197.
- Yourdon, e. (1992) "Análise Estruturada Moderna", tradução da terceira edição, Rio de Janeiro, Campus.