

PRINCIPAIS PROBLEMAS RELACIONADOS AO DESENVOLVIMENTO DE SISTEMAS

Alessandro José Ventorin ¹

RESUMO

O desenvolvimento de sistemas é uma atividade extremamente complexa. Nos últimos anos, houve um crescimento exagerado na indústria de softwares em todo mundo, devido ao avanço e disseminação da informática, o que levou empresas e desenvolvedores a profissionalizarem suas atividades de produção de sistemas, em busca de um melhor produto de software e uma melhor distribuição dos recursos empregados, e conseqüentemente, um aumento em suas vendas e a maximização de seus lucros. Mas nesse período, surgiram também muitos problemas relacionados a esse tipo de atividade, que acabam afetando diretamente a capacidade de produção e os recursos disponíveis em uma empresa de desenvolvimento. Esse estudo teve como objetivo investigar os principais problemas enfrentados pela indústria de software: a demanda reprimida por novos sistemas, a confiabilidade e a manutenção dos sistemas já desenvolvidos. A partir da identificação desses problemas, buscou-se uma solução ou minimização para os mesmos, onde pude concluir que um software desenvolvido obedecendo a um projeto conciso e dentro de um padrão de qualidade pode diminuir as falhas nos sistemas, diminuindo assim, automaticamente, despesas gastas com manutenção e correção de erros, empregando esses recursos na produção de novos sistemas. Este estudo foi realizado através de pesquisa bibliográfica.

PALAVRAS-CHAVE: Software, Análise de sistemas.

ABSTRACT

The development of systems is an extremely complex activity. In the last years, there was an exaggerating growth in the industry of software worldwide, due to the progress and spread of the computer science, what took companies and developers professionalize their activities of production of systems in search of a better software product and a better distribution of the employed resources, and consequently, an increase in their sales and the maximization of their profits. But in that period, it also appeared many problems related to this type of activity that end up affecting the

¹ Tecnólogo em Processamento de Dados, Centro Universitário do Espírito Santo, Professor do Curso de Administração com ênfase em Análise de Sistemas e Coordenador do Núcleo de Processamento de Dados da Faculdade Capixaba de Nova Venécia – UNIVEN.

production capacity and the available resources directly in a development company. This study had as an objective to investigate the main problems faced by the software industry: the demand repressed by new systems, the reliability and the maintenance of the systems developed already. Starting from the identification of those problems, it was looked for a solution or minimization for the same ones, where I could conclude that a developed software obeying a concise project and inside of an ideal quality, it can reduce the flaws in the systems, decreasing like this, automatically, expenses with maintenance and correction of mistakes, using those resources in the production of new systems. This study was done through bibliographical research.

Key-words: software; analise de systems

1 INTRODUÇÃO

Nos dias atuais, o crescimento da tecnologia, a competitividade entre organizações e a busca por informação confiável, precisa e com agilidade, vêm ocasionando um aumento considerável da demanda por novos sistemas computacionais. A busca contínua por essa automação faz com que a demanda tenda a se elevar cada vez mais. A sociedade também exige sistemas cada vez mais sofisticados, o que eleva o grau de dificuldade para quem o está desenvolvendo.

A necessidade de um bom projeto de software passou a ser fundamental para que se tenha o êxito esperado após sua implantação, onde o objetivo maior é satisfazer as necessidades do usuário.

O ciclo de vida de um projeto de software passa pelas fases de identificação do problema e da necessidade do usuário, estudo de viabilidade, análise e projeto de sistema, desenvolvimento, testes e manutenção. Mas, durante esse ciclo de vida, os profissionais e empresas envolvidos enfrentam alguns problemas relativos ao processo de desenvolvimento, e com o passar dos anos esses problemas têm se agravado nesse tipo de atividade.

Esse trabalho foi desenvolvido com o intuito de investigar melhor esses problemas, relatando suas principais causas, e também apresentar soluções para corrigir ou pelo menos minimizar seus efeitos.

2 METODOLOGIA

As informações para realização deste trabalho foram obtidas através de pesquisa bibliográfica, buscando identificar os maiores problemas relacionados ao processo de desenvolvimento de um sistema de informação, e que afetam diretamente seu ciclo de vida.

Partindo desse pressuposto, o estudo foi focado nas principais causas desses problemas, e a que eles estão relacionados.

Por fim, buscou-se uma solução para minimizar tais problemas e apresentá-la aqui como alternativa para melhoria da qualidade do processo de produção de sistemas de informação.

3 DESENVOLVIMENTO

Por se tratar de uma tarefa extremamente complexa, as SDO's (Software Development Organizations), que segundo Rouiller (2003), representam uma organização independente, um departamento ou uma unidade de uma organização, que é responsável por desenvolver, manter, oferecer ou operar um produto ou serviço de software ou sistema de software intensivo, há muitos anos, se deparam com diversos problemas relacionados ao processo de desenvolvimento de sistemas de informação.

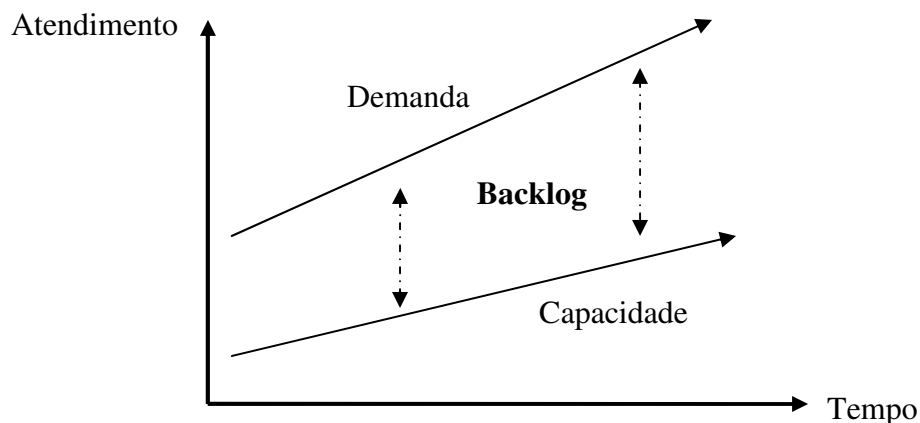
De acordo com Yourdon (1990), os principais problemas no desenvolvimento de sistemas são a produtividade, a confiabilidade e a manutenibilidade.

3.1 PRODUTIVIDADE

A produtividade talvez seja o problema mais evidente enfrentado pelos profissionais de desenvolvimento de sistemas e pelas SDO's, e diz respeito à falta de recursos dessas organizações para atender a uma grande demanda e um mercado exigente.

3.1.1 A DEMANDA REPRIMIDA

O principal causador dos problemas relacionados com a produtividade sem dúvida é a demanda reprimida por novos sistemas, também chamada de Backlog dos aplicativos. Alves (2004) define backlog como uma fila de espera existente na área de sistemas de uma empresa, decorrente do fato de que a demanda por novos sistemas cresce mais depressa que a capacidade da empresa produzir.



*Problema de backlog dos aplicativos
Fonte: Alves (2004)*

De acordo com Yourdon (1990), esse backlog se divide em três tipos: Visível, invisível e desconhecido.

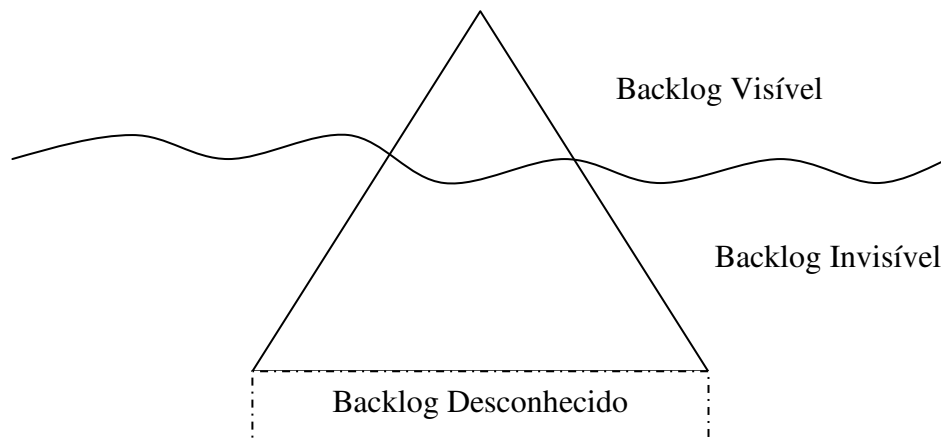
O backlog visível corresponde a novos sistemas solicitados por usuários e que não foram iniciados por falta de recursos das empresas de desenvolvimento, como por exemplo, analistas de sistemas, programadores, equipamentos, etc. Eles estão na fila, esperando que algum outro projeto seja concluído ou que novos recursos sejam adquiridos para que possa ser iniciado o seu desenvolvimento propriamente dito.

O backlog invisível diz respeito à necessidade de novos sistemas, mas que não são solicitados, pois aguardam a conclusão de algum projeto anterior, ou seja, o usuário sabe que precisa do novo sistema, mas ele não irá solicitá-lo oficialmente enquanto o outro não estiver pronto.

O backlog desconhecido corresponde aos novos sistemas que surgirão, tão logo terminem os projetos de backlog visível e invisível e que os usuários nem sabem que precisam.

Yourdon (1990) cita um estudo sobre a demanda reprimida por sistemas de informação feita pelos pesquisadores Robert Alloway e Judith Quillard, da MIT Sloan School, onde foi constatado que o backlog invisível é 5,35 vezes maior que o visível.

Podemos então comparar o problema da demanda reprimida com um iceberg. Enxergamos a ponta do iceberg, apenas uma pequena parte, que é o backlog visível, mas a parte maior, o backlog invisível e desconhecido, permanece oculta sob a água.



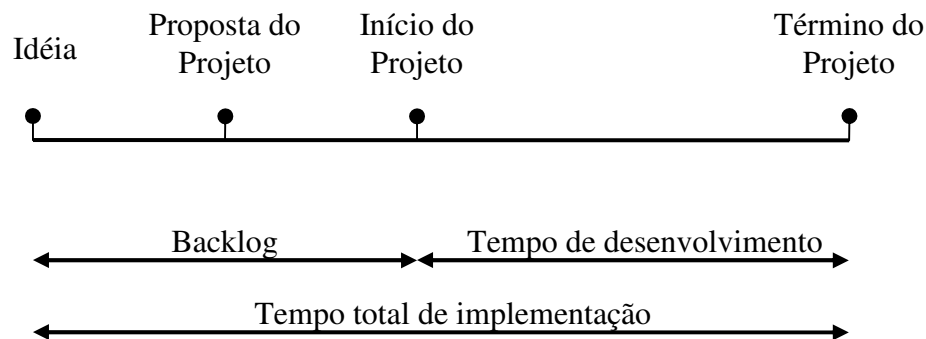
*Analogia do backlog dos aplicativos com um iceberg
Fonte: Alves (2004)*

Essa demanda invisível e desconhecida se torna um problema sério para empresas que preparam seu orçamento baseado apenas na demanda visível, visto que surgirão novos sistemas para serem desenvolvidos cujos gastos para serem projetados não fazem parte de seu orçamento.

3.1.2 O TEMPO DE DESENVOLVIMENTO DE UM SISTEMA

Um segundo aspecto relacionado à produtividade é o tempo gasto para se desenvolver um determinado sistema. Prazos não cumpridos podem implicar em

prejuízos para o usuário, como perdas de oportunidade, além de culminar com um outro problema, que seriam custos acima da previsão por parte do desenvolvedor. Esse outro aspecto está diretamente relacionado com a demanda reprimida, visto que, de acordo com Ambler (1997), o backlog afeta diretamente o tempo necessário para a implementação de um sistema. Podemos dividir esse tempo total de em duas partes: o tempo transcorrido até se iniciar o projeto (backlog) e o tempo necessário para concluí-lo. Isto significa que diminuindo o backlog, poderemos finalizar os projetos mais rapidamente.



*Como o backlog afeta o tempo total para implementação de um sistema.
Fonte: Ambler (1997)*

3.1.3 OS PROJETOS NÃO CONCLUÍDOS

Um terceiro aspecto com relação ao problema de produtividade seriam os projetos que se iniciam más que nunca são terminados, devido à falta de tempo para uma análise adequada, inexperiência da equipe de desenvolvimento, problemas técnicos e gerenciais, falta de recursos (mão-de-obra e equipamentos), dentre outros. Yourdon (1990) afirma que 25% dos projetos em grandes SDO's nunca são concluídos.

As SDO's enfrentam o problema da produtividade há muito tempo. Muitas delas estão procurando soluções e meios de minimizar o problema do backlog e de diminuir o tempo médio gasto para se desenvolver novas aplicações. De acordo com Yourdon (1990) algumas dessas soluções seriam:

- Contratação de mais programadores e analistas de sistemas;
- Contratação de programadores e analistas de sistemas mais talentosos, oferecendo-lhes melhores condições de trabalho;
- Melhores equipamentos e ferramentas de desenvolvimento;
- Aplicação das técnicas de engenharia e qualidade de software;
- Ataque ao problema da manutenção, que veremos especificamente mais adiante, dentre outros.

3.2 CONFIABILIDADE

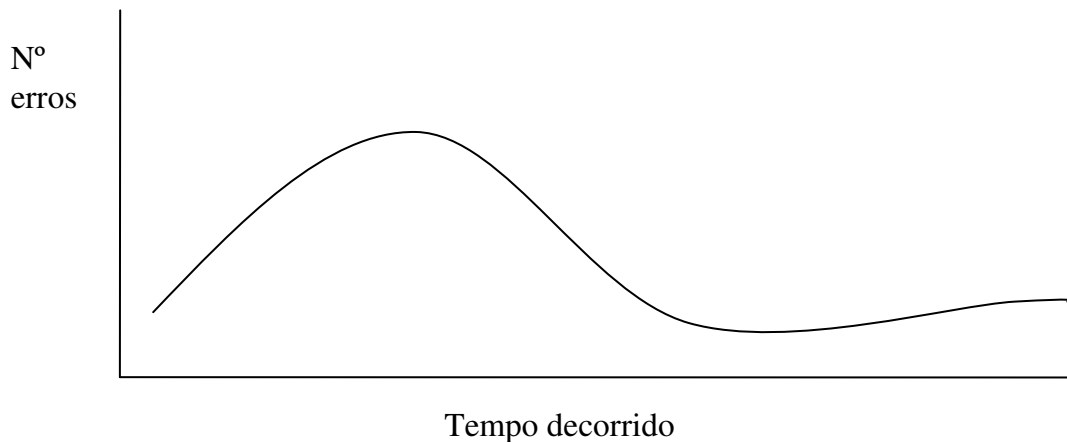
Os erros e falhas que acontecem durante a utilização dos sistemas são causadores do problema da confiabilidade. Sistemas instáveis causam prejuízos como perda de dados, atrasos em atividades importantes, re-trabalho, dentre outros. Conseqüentemente os usuários ficam desconfiados quanto ao sistema e, automaticamente, seu desenvolvedor.

Podemos encontrar erros pequenos, como uma informação impressa incorretamente em um relatório, erros moderados, como um erro de processamento de dados, e erros graves que podem causar prejuízos financeiros ou até perda de vidas humanas.

Yourdon (1990) afirma que, em muitos casos, não existem sistemas totalmente seguros e sem falhas. Alguns erros nunca chegam a serem descobertos, e na maioria deles a documentação é relaxada ou ela nem existe, ficando mais de 50% dos erros sem divulgação, até mesmo dentro das SDO's.

Logo após a implantação do sistema, na liberação para utilização pelo usuário final, a taxa de erros dos sistemas é muito grande, devido à inexperiência dos usuários em

relação à nova ferramenta de trabalho, à fase de testes iniciais, que detecta muitas falhas de programação, ou devido a uma análise incorreta ou mal planejada por parte da equipe de desenvolvimento. Observe o gráfico:



*Erros descobertos em função do tempo de utilização de um sistema
Fonte: Yourdon (1990)*

De acordo com Yourdon (1990), a curva nunca atinge o zero, ou seja, um sistema nunca estará totalmente livre de falhas durante sua utilização. A área sob curva, que representa o número total de erros no início, é muito grande, o que indica um número elevado de erros no período pós-implantação. Outro ponto, é que a curva, algumas vezes, tende a se elevar, marcando períodos de instabilidade do sistema.

Essas falhas não são fáceis de serem corrigidas, pois a equipe de desenvolvimento tem de encontrá-las, e tentar resolvê-las sem que essa correção provoque outros tipos de erros no sistema, como ocorre na maioria das vezes.

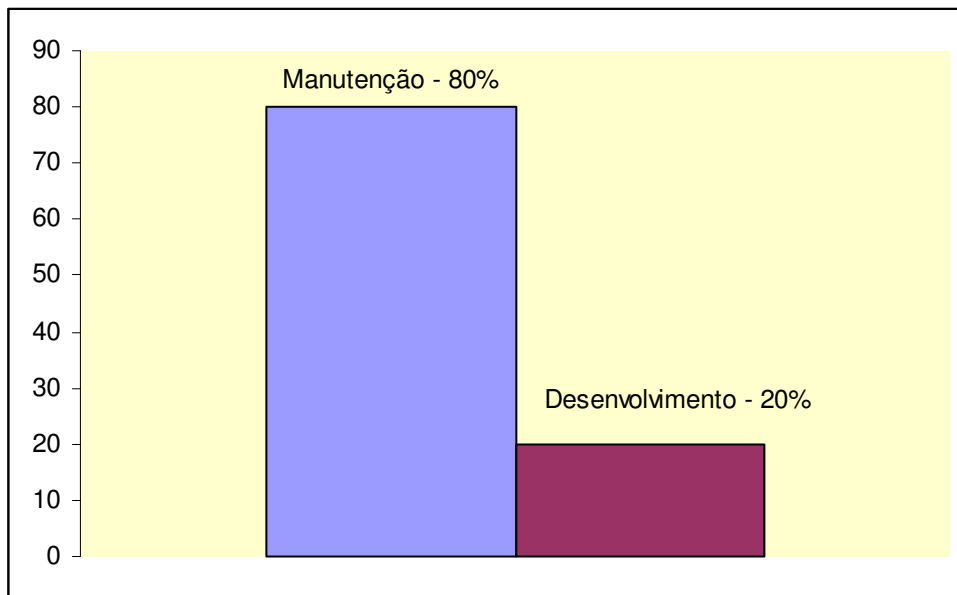
Um bom projeto de análise do sistema, utilizando as novas técnicas de engenharia e qualidade de software ajudaria a reduzir essa grande quantidade de falhas iniciais.

3.3 MANUTENIBILIDADE

A manutenção consiste na modificação, conversão, aperfeiçoamento e correção de falhas dos sistemas desenvolvidos e que se encontram em funcionamento. As SDO's costumam gastar muito dinheiro com essa atividade, utilizando a maioria do tempo e

orçamento com a manutenção de sistemas já desenvolvidos. Esse é o principal causador das filas de projetos (backlog) à espera de serem concluídos.

De acordo com Ambler (1997), o trabalho de manutenção e suporte consome cerca de 80% dos recursos disponíveis de uma empresa de desenvolvimento. Já Yourdon (1990) afirma que a manutenção consome de 50 a 80% de todo trabalho realizado na maior parte das organizações de desenvolvimento de sistemas.



*Alocação típica do orçamento dos departamentos de informática
Fonte: Ambler (1997)*

A correção de erros é um dos aspectos da manutenção. De acordo com Yourdon (1990) eles correspondem a 21% dos trabalhos de manutenção.

Ambler (1997) também cita algumas razões que explicam os gastos elevados com a manutenção:

- O número de sistemas que foram desenvolvidos no passado e que ainda se encontram em uso é muito grande, e assim como você precisa gastar dinheiro na manutenção do seu carro antigo, para continuar rodando, as empresas precisam fazer o mesmo para que os sistemas possam continuar atendendo às suas necessidades.

- A documentação de sistemas é ruim, isto é, quando ela existe, ou seja, em vez de consultarem a documentação para encontrar os erros, os programadores e analistas responsáveis pela manutenção do sistema gastam dias, ou até meses, tentando encontrar o problema no código-fonte do programa.
- Os sistemas antigos são mal construídos, de acordo com os padrões atuais, o que causa uma incompatibilidade de metodologias para manutenção.

Para obter uma diminuição na manutenção dos sistemas, as SDO's precisam maximizar a qualidade dos projetos de desenvolvimento, baseando a produção de sistemas em novas metodologias e novas ferramentas de análise e engenharia de software.

4 CONCLUSÃO

Através deste estudo, pude concluir que, os principais problemas relacionados ao desenvolvimento de sistemas têm uma estreita relação entre eles.

O problema da confiabilidade, causado principalmente por erros e falhas no sistema é o principal causador do problema da manutenção. Muitas falhas causam prejuízos e necessitam de trabalho de manutenção para correção, absorvendo assim a maior parte dos recursos de uma SDO. Por outro lado, a manutenção afeta diretamente o problema da produtividade, pois, os recursos consumidos na manutenção poderiam estar sendo empregados na produção de novos projetos, diminuindo assim a fila de espera dos sistemas, o chamado backlog, e o tempo médio de desenvolvimento de outros projetos.

Isto quer dizer que, com um melhor planejamento e uma melhor qualidade do produto final, podemos diminuir as falhas nos sistemas, diminuindo assim o gasto com a manutenção de erros, e conseqüentemente, empregando esses recursos na produção de novos sistemas, atendendo assim a essa demanda reprimida, maximizando os lucros da organização.

5 REFERÊNCIAS BIBLIOGRÁFICAS

ALVES, Rêmulo Maia; ZAMBALDE, André Luiz; FIGUEIREDO, Cristhiane Xavier. **Sistemas de informação**. Lavras: UFLA/FAEPE, 2004. 79p.

AMBLER, Scott W. **Análise e projeto orientados a objeto**. Rio de Janeiro: Infobook, 1997. 252p.

ROUILLER, Ana Cristina. **Engenharia de software**. Lavras: UFLA/FAEPE, 2003. 126p.

YOURDON, Edward. **Análise estruturada moderna**. 10. ed. Rio de Janeiro: Campus, 1990. 836p.