

# A Torre de Babel e a Qualidade de Software

*Publicado na Developers Magazine – Abril/2002*

*Arisio Costa*

---

No princípio era o programador; e o Programador captava, interpretava, analisava e compreendia o que deveria ser feito; e programava os computadores; e os programas eram bons e os usuários estavam satisfeitos (mesmo porque a tecnologia não permitia muitas variações).

O tempo passou. O mundo evoluiu e a tecnologia, então, evoluiu muito mais rapidamente. Desenvolver e manter em operação os sistemas de informação que atualmente são demandados pelas organizações de médio e grande porte tornou-se uma tarefa extremamente complexa. Os sistemas dos anos 60 cobriam umas poucas atividades auxiliares das empresas, afetando umas poucas pessoas de alguns departamentos. Nesta virada de milênio a maioria das empresas almeja ter todas as suas atividades executadas com o auxílio do computador, sendo que muitas já operam deste modo. Nessas empresas, todos os seus funcionários, fornecedores, clientes, e até mesmo alguns órgãos do governo estão ou estarão de alguma forma envolvidos com seus sistemas.

Estaríamos agora revivenciando o episódio bíblico da Torre de Babel? Conforme nos relata a Bíblia, os homens resolveram construir uma torre cujo cume atingiria o céu, e botaram mãos à obra. À medida que a notícia foi se espalhando, a quantidade de voluntários para construir tal torre - e eventualmente chegar ao céu - foi crescendo assustadoramente. Após algum tempo, a multidão já não se entendia em relação a nada, pois dialetos e línguas diferentes começaram a ser falados por grupos específicos. Uma grande confusão se estabeleceu e o projeto teve que ser abandonado.

O que tem isso a ver com a qualidade de software em pleno início do terceiro milênio? Muito mais do que parece.

É verdade que, enquanto na construção da torre bíblica havia apenas dois materiais (tijolo e piche), a elaboração de qualquer solução de TI é infinitamente mais complexa. Necessitamos de tijolos de diferentes naturezas (como hardware, software básico, middleware, componentes reutilizáveis de software) e que apresentam uma grande variedade de características diferentes entre si, muitas vezes “falando” linguagens e dialetos diferentes. Nosso piche para ligar estes tijolos tecnológicos é o software que implementa as funcionalidades necessárias para a solução a ser disponibilizada. Mas construir software também não se assemelha a extrair e preparar piche.

É ... As diferenças são significativas, mas existe um ponto comum importante: tanto a construção da torre de Babel quanto a construção de um software são projetos únicos. Cada software a ser construído é o equivalente a uma torre específica a ser erigida, com suas características peculiares. Mas, também neste aspecto, o que temos na realidade é uma profunda “dessemelhança”. Podemos imaginar uma ou mais fábricas de tijolos para suprir a construção da torre bíblica. Milhões de unidades idênticas seriam utilizadas. Também podemos imaginar diversas fábricas dedicadas a produzir o piche necessário.

Para um projeto qualquer de construção de software não existe nada semelhante. Os tijolos nunca são iguais e cada um deve ser construído especificamente para ser usado naquele software. Mesmo quando podemos utilizar algum “componente” pré-fabricado isso não é tão simples, e muitas vezes implica em efetuar ajustes em outros tijolos que estamos construindo. Por isso, construir software é amplamente aceito como sendo uma das atividades mais complexas do ser humano, se não a mais complexa. Somente a necessidade de se lidar com uma enorme quantidade de variáveis das mais diversas naturezas, as dependências e os

# A Torre de Babel e a Qualidade de Software

Publicado na *Developers Magazine* – Abril/2002

Arisio Costa

---

relacionamentos entre elas, já proporciona uma ordem de grandeza da complexidade e das possibilidades de falha.

Não sabemos exatamente quando e como os homens falharam na construção da torre desta história<sup>1</sup>. Entretanto, o que me chama a atenção neste episódio é que, mesmo considerando as dificuldades inerentes para uma empreitada desta envergadura naquela época, o que motivou o insucesso não foi a falta de capacidade de construir. Parece até que era suficiente demais e Deus (Javé) teve que interferir, providenciando uma grande dificuldade de comunicação para que o projeto não fosse concluído<sup>2</sup>. Nossos projetos de construção de software estão em situação muito semelhante. São relativamente poucos os projetos que fracassam devido à falta de capacidade técnica para construir. A grande maioria fracassa entregando um produto com características diferentes daquelas esperadas pelo cliente. E isto não acontece em decorrência de erros ou deficiências técnicas, mas sim devido a algum entendimento errôneo do que deveria ser construído, ou seja, dificuldade de comunicação.

É sempre bom lembrar que a qualidade do software que construímos depende visceralmente do bom entendimento dos requisitos reais de nossos clientes (internos ou externos). Um processo definido, conhecido e aceito por todos os envolvidos é importante, na medida que proporciona procedimentos para captar, registrar e validar esses requisitos. Mas não é suficiente. Um processo só é bom quando atinge o objetivo. E qualquer processo desta natureza só atinge seu objetivo com a troca de conhecimento entre pessoas, o que só acontecerá se houver uma comunicação efetiva entre a equipe e todos os “stakeholders”<sup>3</sup> do projeto. Estaremos conseguindo atingir tal objetivo? O quadro anexo – *Chaos Report de 1995*, que sumariza dados de projetos realizados nos EUA – é um indicador de que não.

A comunicação é nosso calcanhar de Aquiles, e, portanto, uma imensa área de oportunidade. Se quisermos dar um salto no nível de qualidade de nossos produtos devemos gerenciar nossos projetos com um foco de atenção especial sobre este assunto. A comunicação é um mecanismo interpessoal totalmente dinâmico. Não pode ser estabelecido ou controlado por um processo, mas podemos incentivar as pessoas a se comunicarem mais efetivamente promovendo um ambiente de trabalho que valorize mais a participação e a colaboração do que a competição. Podemos e devemos fazê-lo, mesmo tendo que envolver outras áreas de nossa empresa. Precisamos fazê-lo, pois a alternativa é nosso projeto se tornar uma pequena Babel e entregar um sistema diferente do esperado ou, transformando-se em uma grande Babel, até mesmo não entregar qualquer sistema.

*Sobre o autor: Arisio Costa é consultor na TI MÉTRICAS e possui ampla experiência em planejamento e execução de projetos de desenvolvimento e manutenção de sistemas, podendo ser contactado em [arisio@gmail.com](mailto:arisio@gmail.com).*

---

<sup>1</sup>Parece que não houve um gerente para o projeto. Também não existe qualquer evidência de um plano de contingências, indicação segura de que os riscos não foram avaliados. Seriam motivos suficientes para o fracasso do projeto? De qualquer forma, não houve uma avaliação ao final do projeto e não temos um registro das causas reais do fracasso.

<sup>2</sup>Então Javé desceu para ver a cidade e a torre que os homens estavam construindo. E Javé disse: “eles são um povo só e falam uma só língua. Isso é apenas o começo de seus empreendimentos. Agora, nenhum projeto será irrealizável para eles. Vamos descer e confundir a língua deles, para que um não entenda a língua do outro.” Gênesis 11-5 a 11-7 - Bíblia Sagrada da Edições Paulinas)

<sup>3</sup>Termo utilizado pelo PMI - Project Management Institute - para designar as pessoas ou entidades que tenham algum tipo de interesse ou possam ser afetadas pelo projeto ou pelo sistema a ser implantado.

# A Torre de Babel e a Qualidade de Software

Publicado na Developers Magazine – Abril/2002

Arisio Costa

## Anexo - *Chaos Report de 1995*

Um estudo denominado “Chaos Report”, publicado pelo Standish Group em 1995 com base em uma ampla pesquisa (mais de 8.000 projetos em mais de 360 empresas nos EUA) sobre projetos completados em 1994, buscou quantificar os sucessos e insucessos e suas causas. A pesquisa classificou os projetos em três tipos:

1. sucesso – projetos que terminaram no prazo, dentro do orçamento e disponibilizaram os requisitos previstos.
2. desafio – projetos que foram concluídos com atraso, com o orçamento “estourado” e/ou não disponibilizaram todos os requisitos previstos.
3. cancelado – projetos que foram cancelados ou cujo produto não foi implantado.

Os resultados não foram muito animadores: apenas 16,2% foram identificados como sucesso, enquanto 52,7% foram considerados desafios e 31,1% foram cancelados.

A pesquisa também identificou os fatores que mais contribuíram para esse enorme contingente (83,8%) de insucessos. A tabela a seguir relaciona os fatores identificados pelos participantes do estudo:

<b>Fatores de Insucesso</b>	<b>% de respostas</b>
1. Falta de participação do usuário	12.8%
2. Requisitos e especificações incompletos	12.3%
3. Mudanças nos Requisitos e especificações	11.8%
4. Falta de suporte gerencial	7.5%
5. Falta de competência tecnológica	7.0%
6. Insuficiência de recursos	6.4%
7. Expectativas não realistas	5.9%
8. Objetivos não definidos com clareza	5.3%
9. Cronogramas não realistas	4.3%
10. Novas tecnologias	3.7%
Outros fatores	23.0%

Embora o estudo não tenha explicitado, pode-se perceber que algum tipo de deficiência na comunicação deve ser a causa real por trás de muitos dos fatores de insucesso acima.

Não dispomos de dados sobre o mercado brasileiro, mas parece-me razoável admitir que estejamos em situação semelhante.