

---

## **PLoPime 2006**

### **Anti-Padrão de Desenvolvimento:**

### **Desastre Incomensurável**

**Autores: Mario Marques Junior e Reginaldo Rideaki Kamiya**

---

*incomensurável* adj. m. e f. 1. Não comensurável; imensurável. 2. Mat. Sem medida comum com outra grandeza.

#### ***Uma anedota para ilustrar...***

Jean Claude: — *Ei, John. Estou fazendo o monitoramento da sonda espacial e temos um problema. Os dados da velocidade não estão batendo, e a sonda está completamente fora da rota planejada.*

John Smith: — *Absolutamente não, Jean Claude. Nas minhas observações, todos os dados de velocidade, desde o lançamento, estão de acordo com o planejado.*

Jean Claude: — *Não entendo... Em seus dados, qual a velocidade atual da sonda?*

John Smith: — *138 pés/s, com desvio de 0,5 pés/s e probabilidade 0,99.*

Jean Claude: — *M\*! 138 PÉS POR SEGUNDO? Não deveriam ser METROS? Argh!! Já é tarde. Perdemos o contato com a sonda!! ...*

#### ***Raízes do Problema***

Uma falha no projeto, por incoerência no uso de unidades de medida; usualmente em sistemas em que se espera alta confiabilidade e têm alto custo; é encontrado apenas quando ele já está em uso.

#### ***Exemplos***

##### **Mars Climate Orbiter (MCO)<sup>1</sup> (Setembro/1999)**

Essa sonda espacial, juntamente com Mars Polar Lander, fez parte do programa Mars Surveyor '98 da NASA, para estudo da temperatura e condições climáticas do planeta Marte, além do levantamento do montante de água e dióxido de carbono em sua superfície, com a finalidade de compreender o comportamento e as regras de mudanças atmosféricas.

O programa teve como resultado a baixa das duas sondas, porém no caso da nave Mars Climate Orbiter em especial a falha foi relacionada ao mau uso de unidades de medida. Foi descoberto que alguns dados eram calculados em Unidades Imperiais (pound força – segundo) e passadas dessa forma para a equipe de controle de navegação, enquanto que essa esperava os dados em Unidades Métricas (newton – segundo). Isso fez com que a sonda ao invés de entrar na atmosfera de Marte na

altitude prevista de 140–150km, o fizesse na altitude de 57km, sendo destruída devido ao estresse atmosférico e atrito superiores a baixas altitudes. A sonda não tinha capacidade de fazer conversões entre os dois sistemas de medida.

Fatores que agravaram esse erro foram a falta de testes relativos a De-saturação de Momento Angular (ou *Angular Momentum Desaturation* – AMD). Estes testes seriam necessários tanto antes do lançamento, com a verificação dos fluxos de dados provenientes do algoritmo baseado em AMD; quanto em vôo, com um algoritmo de navegação independente para verificação cruzada. Ambos testes já estavam sendo usados em outras missões, mas foram cortados dessa por questões de orçamento.

### **Gimli Glider<sup>2</sup> (Julho/1983)**

Esse é um apelido dado a um jato Boeing 767-200 da Air Canada, que teve pane seca a 12000 metros de altitude, e aproximadamente na metade do percurso entre Montreal na região leste do Canadá; e Edmonton, na região oeste do mesmo país. O motivo do nome foi devido à forma com que o avião aterrissou, deslizando, na região de Gimli, em uma antiga base aérea.

Um jato Boeing 767 é abastecido usando-se um aparelho chamado *Fuel Quantity Information System Processor*, que é responsável por operar todas as bombas de abastecimento e reportar aos pilotos as informações sobre a carga de combustível. Para esse vôo, ele estava quebrado, forçando a tripulação a medir manualmente o volume nos tanques, com um *dripstick*. Porém os cálculos para a quantidade de combustível necessário eram feitos em massa, sendo necessário fazer uma conversão nas medidas. O erro ocorreu porque o avião em questão era o primeiro da Air Canada que media combustível em quilogramas ao invés de pounds. Com isso, um fator de conversão errado acarretou no abastecimento insuficiente, ocasionando na queda.

### **Sintomas e Conseqüências**

Os sintomas são a falta de comunicação no projeto, além de interpretação diferente dos dados. Esse último sintoma é geralmente notado tão tardiamente que leva o sistema a uma morte súbita. A conseqüência imediata desse descuido é a falência do projeto como um todo, e desperdício de recursos, pois geralmente esse tipo de projeto demanda um alto orçamento, tanto em hardware quanto em profissionais capacitados. A conseqüência seguinte é gerar um descrédito em sistemas desse tipo, fazendo com que seja muito mais difícil aprovar um novo projeto desse porte; a não ser que se invista em testes, o que torna o projeto muito mais caro. Apesar desse investimento, o Desastre Incomensurável continua a ocorrer, fazendo com que o orçamento para testes suba ainda mais, inviabilizando uma nova empreitada. É uma bola de neve.

### **Causas Típicas**

Existem duas fontes de problemas que podem causar esse Anti-Padrão. Em primeiro lugar, porque os desenvolvedores continuam usando unidades de medida que não são compatíveis? Possivelmente:

- Esse tipo de projeto envolve desenvolvedores de vários países. Um desenvolvedor de um determinado país pode se sentir mais à vontade

trabalhando com unidades e constantes familiares, principalmente em magnitudes que ele consegue ver/sentir.

- As unidades usadas não são consenso entre os desenvolvedores, principalmente quando foi decidida por superiores e há falhas de comunicação.

Depois, porque os defeitos continuam no sistema, mesmo se investindo em testes? Pode ser que:

- Os desenvolvedores testam seus componentes isoladamente, e não há um teste que envolva todos os componentes do sistema, ou;
- Mesmo havendo um teste de integração, ele também é incoerente em relação às unidades de medida.

### ***Solução Refatorada***

- Deixe o time entrar em acordo sobre quais as unidades e constantes usadas no sistema. Para equipes em um mesmo local, use quadro negro para ter visualização constante do que foi fixado e facilitar a comunicação. Para equipes distribuídas, use um quadro virtual, mas tenha certeza que todos o consultam freqüentemente.
- Uma alternativa à solução acima é definir um conjunto de conversões corretas, ao invés de unidades padronizadas. Porém o sistema de conversões é dependente de testes e manutenção próprios.
- Tenha uma equipe centralizada para desenvolver testes de integração. Isso não impede os testes de unidade a priori, ou seja, testes sobre os componentes locais feitos pela própria equipe desenvolvedora, que podem ter unidades incoerentes com as unidades padronizadas enquanto não forem integradas.

### ***Soluções e Anti-Padrões Relacionados***

- A falta de testes que causou a falha de sonda MCO provavelmente pode ter sido motivada pelo Anti-Padrão Blind Faith<sup>3</sup> (fé cega).
- A construção e uso de Glossários<sup>4</sup> no sistema é uma solução semelhante. A convenção das unidades de medida e conversões podem tanto fazer parte de um glossário geral do sistema quanto estar em um artefato à parte.

### ***Referências***

[1] URL: [http://sunnyday.mit.edu/accidents/MCO\\_report.pdf](http://sunnyday.mit.edu/accidents/MCO_report.pdf)

[2] URL: <http://www.casa.gov.au/avreg/fsa/03jul/22-27.pdf>.

[3] URL: [http://en.wikipedia.org/wiki/Blind\\_faith\\_\(computer\\_science\)](http://en.wikipedia.org/wiki/Blind_faith_(computer_science)).

[4] Scott W. Ambler, The Object Primer 3<sup>rd</sup> Edition, Agile Model Driven Development with UML2, Cambridge University Press, 2004.