UNIVERSIDADE FEDERAL DE PERNAMBUCO
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO
CENTRO DE INFORMÁTICA
2008.2

Padrões para Implantação de Programas de Medição em Organizações de Software

Qualidade, Processos e Gestão de Software

Nome: Renata de Avelar Alchorne (raa3@cin.ufpe.br)
Professor: Alexandre Marcos Lins de Vasconcelos (amlv@cin.ufpe.br)

26 de Janeiro de 2009

Resumo

Este trabalho tem por objetivo apresentar os conceitos relacionados a Padrões que estão ligados à produção de software em organizações. Através disto, introduzir a idéia de Padrões para Implantação de Programas de Medição nas Organizações produtoras de software, tendo em vista que as atividades de implantação deste tipo de programa enfrentam diversas dificuldades, principalmente no que diz respeito à resistência imposta pela cultura da organização. Observa-se que estas dificuldades estão sendo solucionadas de forma semelhante nas empresas fazendo com que surja um modelo a ser seguindo para desempenhar tais atividades a fim de atingir uma implantação com êxito. A esta descriçãom dá-se o nome de Padrão, que se trata de um par problema-solução em um determinado contexto com o intuito de obter sucesso na execução de uma determinada atividade.

Conteúdo

Lista de Figuras

1. Introdução

Desde os anos 90, a noção de reuso de software tem sido adotada no ciclo de desenvolvimento de software [1] e esta prática está sendo cada vez mais aprimorada a fim de tornar a produção de software mais lucrativa através da redução de esforço e custo proporcianados pela reutilização de componentes de software.

A idéia por trás do reuso é reaproveitar grande parte de um software que foi produzido em projetos futuros a fim de obter as vantagens citadas anteriormente. O esforço de produção ocorrerá apenas uma única vez e a reutilização de partes do software, aliados a dados históricos de projetos anteriores permite a aplicação em novos sistemas desenvolvidos.

Juntamente com o surgimento do paradigma orientado a objetos, através da linguagem Smalltalk, no início dos anos 60 [2], houve um grande entusiasmo na reutilização de software. Dentro desta perspectiva, os padrões de software têm sido pesquisados na última década como uma das forma de reuso e não se restringindo apenas aos famosos padrões de projeto, mas também na área de análise, arquitetura e do próprio processo de desenvolvimento.

Um padrão é conhecido, basicamente, como sendo um par "problema-solução" que tem como objetivo final servir a desenvolvedores e outros membros de projetos, que ao enfretarem problema similares ao apresentado no padrão, possam utilizá-lo de forma a resolver o problema em questão.

2. Padrões

Os primeiros conceitos de padrões surgiram na área de arquitetura, através dos trabalhos de Christopher Alexsander, nas décadas de 60 e 70. Alexsander criou as primeiras definições para os termos padrão e linguagens de padrões (que será detalhada mais adiante). Estes termos também são comumente referenciados na área de software por se aplicarem ao domínio da engenharia de software [3].

Algumas definições foram sendo dadas para padrões, como as destacadas abaixo:

"Cada padrão é uma regra de três partes que expressa uma relação entre um certo contexto, um problema e uma solução" (Alexander, 1979). [4]

"Um padrão é uma entidade que descreve um problema que ocorre repetidamente em um ambiente e então descreve a essência de uma solução para este problema, de tal forma que você possa usar essa solução milhões de vezes, sem nunca utilizá-la do mesmo modo" (Alexander et al., 1977). [5]

"Como um elemento no mundo, cada padrão é uma relação entre um certo contexto, um certo sistema de forças que ocorrem repetidamente naquele contexto, e uma certa configuração espacial que permite que estas forças se solucionem por si só" (Alexander, 1979). [4]

Seguindo o mesmo raciocínio dos padrões da arquitetura, a engenharia de software tem apresentado boas soluções para problemas conhecidos nas diversas áreas deste domínio: análise, projeto, arquitetura, interface, acesso a dados e processos de desenvolvimento. James Coplien descreveu um padrão de software da seguinte forma [3]:

"Um padrão é um pedaço de literatura que descreve um problema de projeto e uma solução geral para o problema num contexto particular" (Coplien, 1996). [6]

Considerando tudo que foi mostrado até agora, podemos dizer que um padrão é um par do tipo problema-solução, que documenta uma dada solução para um problema recorrente apresentando o contexto sob o qual o tal problema ocorre e a solução funciona. Neste trabalho, consideraremos esta a definição de padrão.

Christopher Alexsander também estabeleceu que os mesmos devem ter as seguintes características que são apresentadas a seguir, por considerarmos relevantes no âmbito dos padrões na engenharia de software [7]:

- Encapsulamento: Um padrão encapsula um problema-solução bem definido. Ele deve ser independente, específico e formulado de maneira a ficar claro onde ele se aplica.
- Generalidade: Todo padrão deve permitir a construção de outras realizações a partir deste padrão.
- Equilíbrio: Quando um padrão é utilizado em uma aplicação, o equilíbrio dá a razão, relacionada com cada uma das restrições envolvidas, para cada passo do projeto. Uma análise racional que envolva uma abstração de dados empíricos, uma observação da aplicação de padrões em artefatos tradicionais, uma série convincente de exemplos e uma análise de soluções ruins ou fracassadas pode ser a forma de encontrar este equilíbrio.
- Abstração: Os padrões representam abstrações da experiência empírica ou do conhecimento cotidiano.

- Abertura: Um padrão deve permitir a sua extensão para níveis mais baixos de detalhes.
- Combinatoriedade: Os padrões são relacionados hierarquicamente. Padrões de alto nível podem ser compostos ou relacionados com padrões que endereçam problemas de nível mais baixo.

Em 1987, a partir dos conceitos criados por Alexsander, os primeiros padrões relacionados a projeto de software foram sugindo através dos programadores Kent Bech e Ward Cunningham, que apresentaram padrões para construção de janelas na linguagem Smalltalk. Mais tarde, outros seguidores foram definindo padrões para cada domínio, como foi o caso dos quatro autores do livro *Design Patterns of Reusable Object-Oriented Software*, conhecidos como a "Gangue dos Quatro" (Gang of Four) ou simplesmente "GoF". Após o GoF, muitos outros padrões surgiram dentro do âmbito de projeto [7].

Mesmo com todo o embasamento e características apresentadas, vale salientar que a comunidade de padrões de software tem o objetivo de documentar e compartilhar soluções comprovadas de engenheiros de software experientes para problemas recorrentes em um determinado contexto, criando assim uma literatura que auxilie na adoção das boas práticas para o desenvolvimento de sistemas. Vlissides [3] cita quatro benefícios do reuso de padrões:

- Capturar experiências, tornando-as acessíveis aos não-experientes;
- Formar um vocabulário a fim de ajudar desenvolvedores a se comunicarem melhor;
- Ajudar engenheiros de software a entender um sistema mais rapidamente quando ele está documentado com os padrões reutilizados;
- Facilitar a reestruturação de um sistema, tendo ele sido ou não projetado com padrões em mente.

Além dos benefícios listados, foi realizada uma pesquisa [8] onde são apresentadas outras vantagens da utilização de padrões de projeto, especificamente, no desenvolvimento de sistemas:

- Evolução de código;
- Modularidade;
- Desacoplamento entre áreas de responsabilidades de forma que as mudanças em uma área não ocasionem mudanças nas outras;
- Diminuição da complexidade do projeto e do código final;
- Facilidade na criação de frameworks contendo padrões de projeto já implementados a fim de facilitar o reuso dos padrões posteriormente;
- Estabilidade do código;

- Confiabilidade no reuso de padrões cujas soluções são comprovadas;
- Ganho de produtividade;
- Facilidade de repassar conhecimento entre os desenvolvedores experientes; e
- Facilidade de aprendizado de novas áreas de conhecimento para uma equipe sem experiência na aplicação a ser desenvolvida.

Desvantagens também existem na utilização de padrões. Algumas delas foram apontadas em casos particulares de padrões voltados para arquitetura, tais como [3]:

- Perda de eficiência: O uso de alguns padrões pode tornar um programa mais lento, uma vez que às vezes é preciso adicionar novas classes ou novas camadas de aplicação ao modelo original do sistema, o que pode causar um retardo na sua execução;
- Diminuição da Legibilidade/Manutenibilidade: Um padrão pode diminuir a compreensão de um projeto ou de uma implementação. A aplicação de um padrão pode ocasionar em um aumento do número de classes, mensagens, linhas de código, níveis hierárquicos de classes, tornando o código do programa menos legível em virtude do aumento da complexidade do código, levando a um custo maior de manutenção, principalmente para programadores que não tenham algum conhecimento técnico sobre os padrões aplicados;
- Falta de Motivação da Equipe: Quando uma equipe de desenvolvimento não tem um entendimento prévio da solução de um padrão, das consequências da sua aplicação e do relacionamento com outros padrões, os membros desta equipe podem se tornar desmotivados quanto à aplicação deste padrão.

2.1 Anti-padrões

Uma vez que os padrões "ditam" através de uma boa prática o que deve ser feito para solucionar um determinado problema conhecido dentro de um domínio, os anti-padrões vêm para mostrar lições aprendidas sobre coisas que não devem ser feitas. Os anti-padrões podem ser de dois tipos: aqueles que descrevem uma solução ruim para um problema que resultou em uma situação ruim e aqueles que descrevem como escapar de uma situação ruim e como proceder para, a partir dela, atingir uma boa solução.

Os anti-padrões são importantes, pois é tão relevante ver e entender soluções ruins quanto boas. Isto é, deve-se mostrar também o que não deu certo e que deve ficar como lição aprendida. [9]

2.2 Estrutura de um Padrão

Ainda no trabalho do arquiteto Christopher Alexsander, foi definido um formato que um padrão deve ter. Basicamente, foram propostos cinco elementos básicos para tal estrutura [7]: Nome, Exemplo, Contexto, Problema e Solução. Além dessa estrutura básica, também foram incorporados outros elementos que foram considerados importantes de acordo com a literatura [9] e que serão apresentados abaixo como sugestão para este trabalho.

- Nome: Uma descrição da solução, mais do que do problema ou do contexto. Todo padrão deve ter um nome significativo. Pode ser uma única palavra ou frase curta que se refira ao padrão e ao conhecimento ou estrutura descritos por ele. Se o padrão possuir mais do que um nome comumente usado ou reconhecível na literatura, subseções "*Aliases*" ou "*Also know as*" devem ser criadas.
- Contexto: A descrição das situações sob as quais o padrão se aplica. Trata-se de pré-condições dentro das quais o problema e sua solução constumam ocorrer e para as quais a solução é desejável, o que reflete a aplicabilidade do padrão. Pode também ser considerado como a configuração inicial do sistema antes da aplicação do padrão.
- Problema: Estabelece o problema a ser resolvido pelo padrão, descreve a intenção e objetivos do padrão perante o contexto e forças específicas.
- Forças: Descrição dos impactos, influências e restrições relevantes para o problema e de como eles interagem ou são conflitantes entre si e com os objetivos a alcançar. Um cenário concreto que serve como motivação para o padrão.
- Solução: Relacionamentos estáticos e regras dinâmicas, descrevendo como construir métricas e implementar práticas de acordo com o padrão, a fim de obter o resultado desejado. Frequentemente citando variações e formas de ajustar a solução segundo as circunstâncias. Inclui referências a outras soluções e o relacionamento com outros padrões de nível mais baixo ou mais alto. Permite as seguintes subseções:
    - "*Structure*", revelando a forma a organização estática do padrão;
    - "*Participants*", descrevendo cada um desses componentes;
    - "*Dynamics*", exibindo o comportamento dinâmico do padrão;
    - "*Implementation*", mostrando detalhes de implementação do padrão;
    - "*Variants*", discutindo possíveis variações e especializações da solução.
- Exemplo: Uma ou mais aplicações do padrão que ilustram, num contexto inicial específico, como o padrão é aplicado e transforma aquele contexto em contexto final.

Figuras, diagramas, gráficos ou descrições auxiliam e ilustram a prática do programa de medição.

- Resultado: O estado ou configuração do sistema após a aplicação do padrão, podendo ter uma subseção "*Consequências*" (tanto boas quanto ruins). Descreve as pós-condições e efeitos colaterais do padrão.
- Passo-a-passo (raciocínio): Uma explicação das regras ou passos do padrão que explicam como e porque ele trata suas influências contrárias, definidas em "Forças", para alcançar os objetivos, princípios e filosofia propostos. Isso nos diz realmente como o padrão funciona, porque funciona e porque ele é bom.
- Padrões relacionados: Os relacionamentos estáticos e dinâmicos desse padrão com outros dentro da mesma linguagem ou sistema de padrões. Padrões relacionados geralmente compartilham as mesmas influências. São possíveis os seguintes tipos de padrões relacionados:
    - Padrões predecessores, cuja aplicação conduza a esse padrão;
    - Padrões sucessores, que devem ser aplicados após esse;
    - Padrões alternativos, que descrevem uma solução diferente para o mesmo problema, mas diante de influências e restrições diferentes;
    - Padrões codependentes, que podem (ou devem) ser aplicados simultaneamente com esse padrão.
- Uso conhecido: Descreve ocorrências conhecidas do padrão e sua aplicação em sistemas existentes. Isso ajuda a validar o padrão, verificando se ele é realmente uma solução provada para um problema recorrente.

2.3 Classificação de Padrões

Os padrões de software abrangem diferentes níveis de abstração, podendo portanto ser classificados em diversas categorias de modo a facilitar sua recuperação e uso. Porém, essa classificação não é rigorosa, podendo haver padrões que se encaixem em mais de uma categoria. Abaixo são apresentadas algumas categorias importantes de padrões [9].

- Padrões de processo: Definem soluções para os problemas encontrados nos processos envolvidos na engenharia de software – desenvolvimento, controle de configuração, testes, etc.

- Padrões arquiteturais: Expressam o esquema ou organização estrutural fundamental de sistemas de software ou hardware. Dentro desta categoria, podemos citar os padrões MVC (*Model-View-Controller*).
- Padrões de padrão: Conhecido em inglês como *Patterns on patterns*, são padrões descrevendo como um padrão deve ser escrito, ou seja, que padroniza a forma com que os padrões são apresentados aos seus usuários.
- Padrões de análise: Descrevem soluções para problemas de análise de sistemas, embutindo conhecimento sobre o domínio de aplicação específico.
- Padrões de projeto: Definem soluções para problemas de projeto de software. Essa categoria de padrões é uma das mais conhecidas para pessoas com formação em computação que trabalham diretamente com desenvolvimento. Ex.: Visitor (muito utilizado na implementação de compiladores).
- Padrões de interface: Definem soluções para problemas comuns no projeto da interface de sistemas.
- Padrões de programação: Descrevem soluções de problemas particulares de uma determinada linguagem ou estilos de programação. Ex.: Padrão de codificação Java.
- Padrões de persistência: Descrevem soluções para problemas de armazenamento de informações em arquivos ou banco de dados. Ex.: XML.

3. Coletâneas de Padrões

Os padrões têm sido propostos por diversos autores nas mais diversas áreas de aplicação. Um padrão não é imposto ou inventado, ele é descoberto após a abstração de fatores comuns em diversos pares problema-solução. Existem os padrões de domínio bem específico, como padrões para hipermídia, e também padrões genéricos, independentes de domínio, como os padrões de interface [9].

Apesar da proposta do padrão ser a de solucionar um problema recorrente em um dado domínio, é importante ressaltar que a aplicação de um determinado padrão pode gerar outros problemas que podem, por sua vez, ser resolvidos através da aplicação de outros padrões. Em geral, não existem padrões isolados: um padrão pode depender de outro no qual esteja contido, ou das partes que ele contém, ou ser uma variação de outro, ou ser uma combinação de outros. De modo geral, um padrão e seus variantes descrevem soluções para problemas muito similares, que variam em algumas das influências envolvidas.

Dessa forma, pensou-se em agrupar os padrões segundo alguns critérios com o objetivo de recuperá-lo e reusá-lo mais facilmente. Isso pode ser feito através das coletâneas de padrões que serão detalhadas mais adiante.

3.1 Coleções de Padrões

Uma coleção de padrões é uma coletânea qualquer de padrões que não possuem nenhum vínculo entre si e, em geral, nenhuma padronização no formato de apresentação. Os padrões podem estar reunidos por terem sido apresentados em um mesmo congresso, por terem sido propostos pelo mesmo autor, ou por se referirem a um mesmo domínio, mas não possuem um relacionamento semântico significativo.

3.2 Catálogos de Padrões

Um catálogo de padrões é uma coletânea de padrões relacionados (talvez relacionados apenas fracamente ou informalmente). Em geral, subdivide os padrões em um pequeno número de categorias abrangentes e pode incluir algumas referências cruzadas entre padrões [9] [10].

Um catálogo de padrões pode oferecer um esquema de classificação um esquema de classificação e recuperação de seus padrões, já que eles estão subdivididos em categorias. Um catálogo de padrões adiciona uma certa quantidade de estrutura e organização a uma coleção de padrões, mas usualmente não vai além de mostrar apenas as estruturas e relações mais superficiais [9] [10].

3.3 Sistemas de Padrões

Um sistema de padrões é um conjunto coeso de padrões co-relacionados que trabalham juntos para apoiar a construção e evolução de arquiteturas completas. Além de ser organizado em grupos e subgrupos relacionados em múltiplos níveis de granularidade, um sistema de padrões descreve as diversas inter-relações entre os padrões e seus grupamentos e como eles podem ser combinados e compostos para resolver problemas mais complexos. Os padrões num sistema de padrões devem ser descritos num estilo consistente e uniforme e precisam

cobrir uma base de problemas e soluções suficientemente abrangente para permitir a construção de parcelas significativas de arquiteturas completas [9] [10].

Um sistema de padrões interliga padrões individuais, descreve como os padrões que o constituem estão conectados com outros padrões no sistema, como esses padrões podem ser implementados e como o desenvolvimento de software com padrões é apoiado. Um sistema de padrões é um veículo poderoso para expressar e construir arquiteturas de software [9] [11].

Além da estrutura e organização oferecidas por um catálogo de padrões, num sistema de padrões é adicionada uma maior profundidade à estrutura, maior riqueza à interação dos padrões e maior uniformidade ao catálogo de padrões [9] [10].

3.4 Linguagens de Padrões

Uma linguagem de padrões é uma coleção estruturada de padrões que se apóiam uns nos outros para transformar requisitos e restrições numa arquitetura [12].

Os padrões que constituem uma linguagem de padrões cobrem todos os aspectos importantes em um dado domínio. Pelo menos um padrão deve estar disponível para cada aspecto da construção e implementação de um sistema de software: não pode haver "vazios" ou "brancos". Por isso, trata-se de um trabalho mais cuidadoso, já que todas as lacunas conhecidas do dado domínio precisam ser preenchidas por soluções propostas pelo padrão.

Uma linguagem de padrões é uma forma de subdividir um problema geral e sua solução complexa em um número de problemas relacionados e suas respectivas soluções. Cada padrão da linguagem resolve um problema específico no contexto comum compartilhado pela linguagem. É importante notar que cada padrão pode ser usado separadamente ou com um certo número de padrões da linguagem. Isso significa que um padrão sozinho é considerado útil mesmo se a linguagem não for ser usada em sua plenitude [9].

4. Padrões para Implantação de Programas de Medição

Sabe-se, atualmente, que um dos maiores desafios das organizações de desenvolvimento de software é medir a produtividade dos seus funcionários a fim de atingir um custo ideal na produção de software, maximizando o lucro da organização. Para isso, diversos estudos têm sido desenvolvidos para entender quais são as formas mais adequadas de se fazer essas medições sem que o trabalho desenvolvido pelos funcionários seja afetado pela realização das mesmas.

Para isso, este trabalho apresenta, nesta seção, a proposta de um conjunto de padrões para implantação de programas de medição em organizações de software. Esses padrões são propostos com base na observação das melhores práticas adotadas em duas empresas que possuem desenvolvimento de software de grande porte na cidade de Recife, estado de Pernambuco.

Com base nas estruturas de padrões vistas nas seções anteriores, adotou-se a seguinte estrutura para a representação dos pares problema-solução da implantação de programas de medição: Nome, Contexto, Problema, Solução, Exemplo e Padrões relacionados. Trata-se de uma estrutura um pouco diferente da apresentada, porém que parece adequada ao domínio em questão. Foram selecionados 10 padrões, dentre os quais, alguns não possuem o componente "Exemplo", mas que em estudos futuros serão trabalhados neste aspecto de forma a apresentar valores para todos os componentes de sua estrutura. [17]

Padrão 1: Alinhar-se com a alta gerência

Contexto:

Em uma organização, a alta gerência contém os principais *stakeholders* para análise e tomada de decisão em cima dos resultados das medições realizadas.

Problema:

- A alta gerência se opõe às métricas;
- A alta gerência não requer explicitamente os dados das métricas. Apenas quando essas métricas são apresentadas, é que a alta gerência critica ou sugere mudanças para as mesmas;
- A alta gerência não participa diretamente da implantação do programa de medição.

Solução:

- Alinhar o programa de métricas com os objetivos de negócio da organização, envolvendo a alta gerência no processo de definição das métricas no nível estratégico;
- Envolver a alta gerência no processo de implantação do programa de medição.

Exemplo:

- A alta gerência define os objetivos de medição ou demanda, explicitamente, informações necessárias;

- As medições são derivadas dos objetivos ou necessidades de informações (aplicação do GQM);
- A alta gerência aprova as métricas definidas;
- A alta gerência participa da análise dos indicadores, analisando os resultados e planejando as ações decorrentes da análise, como tomadas de decisão.

Padrões relacionados:

- Nenhum

Padrão 2: Evitar excesso de métricas

Contexto:

Em uma organização de software, muitos atributos podem ser medidos, como processos, produtos, projetos e recursos.

Diferentes *stakeholders* requerem diferentes conjuntos de informações, que atenda as suas necessidades específicas dentro do domínio de atuação.

Problema:

- Iniciar o programa de medição com muitas métricas, provocando grande esforço de coleta das métricas bem como impossibilidade de analisar todos os dados obtidos.

Solução:

- Iniciar o programa com um conjunto pequeno de métricas, de forma balanceada, como uma métrica por área (que a represente), por exemplo;
- Expandir o conjunto de métricas à medida que os participantes se habituem com a coleta das métricas, análise das mesmas e percebam o valor das informações.

Exemplo:

- Elaboração de um cronograma de implantação do programa de medição bem como sua expansão ao longo de um período.

Padrões relacionados:

- Padrão 1

Padrão 3: Evitar excassez de métricas

Contexto:

Na tentativa de evitar o problema relacionado ao Padrão 2, inicia-se o programa com pouquíssimas métricas.

Problema:

- Não provê informação suficiente para tomada de decisão;
- Pode levar à conclusão que o esforço do programa não vale a pena, já que as informações obtidas são mínimas e não respondem aos questionamentos da área ou alta gerência.

Solução:

- Realizar medições de aspectos complementares para a formação sólida da informação (ex.: Produtividade associada à Qualidade);
- Priorizar segundo a importância das medições;
- Distribuir as medições de forma a não sobrecarregar um papel ou pessoa.

Exemplo:

- Primeira versão do programa de medição tinha indicadores relacionados às seguintes facetas:
    - Gestão do Projeto
        - Entregas no prazo
        - Variação do esforço
        - Variação do custo
        - Variação da receita
    - Qualidade do Projeto
        - Aderência ao processo
        - Resolução de desvios de processos
        - Nota de avaliação de satisfação do cliente
- Foram priorizadas as medições que já eram realizadas de alguma forma.

Padrões relacionados:

- Padrão 1
- Padrão 2

Padrão 4: Medir aspectos corretos

Contexto:

Muitos atributos, em organizações de software, podem ser medidos. Há uma tendência natural de se medir o que é mais simples ou apenas o que é requerido pela alta gerência.

Problema:

- Os itens coletados não estão diretamente relacionados à estratégia de negócio da organização;
- Os gerentes não estão obtendo informações que eles precisam pra gerenciar melhor suas equipes;
- Não se consegue avaliar se mudanças ocorridas no processo trouxeram melhorias.

Solução:

- Selecionar métricas que ajudarão a avaliar as melhorias de processo;
- Utilizar GQM ou outros métodos similares.

Exemplo:

- Não possui

Padrões relacionados:

- Padrão 1

Padrão 5: Definir precisamente as medições

Contexto:

Existe uma tendência natural em priorizar a geração do gráfico, antes mesmo de se pensar ou se definir precisamente o que significa a medição.

Problema:

- Definição vaga ou ambígua da métrica, gerando várias interpretações de um mesmo dado por diferentes indivíduos, além de retrabalho e dificuldades na coleta das métricas.

Solução:

- Definir, da forma mais completa e precisa possível, o conjunto de métricas que devem ser utilizadas no programa de medição.

Exemplo:

	Entregas no Prazo	
Definição	**Sigla:**	**OTD**
	Descrição:	Mostra qual o percentual de entregas atendidas no prazo em relação ao total de entregas planejadas no período
	Fórmula	**OTD=(QTD_ENT_PRAZO/QTD_ENT_PLAN)**
	Procedimento de cálculo	**Deve-se calcular a razão entre as duas métricas básicas, conforme fórmula acima, no período corrente.**
	Unidade Padrão	**%**
	Meta	**100%**
	Escopo	**Projeto**
Análise	**Forma de Apresentação**	**Gráfico 1**: Gráfico em linhas e coluna, apresentando o valor do indicador em linha em cada período de análise, juntamente com os valores das métricas básicas em colunas (plotadas no eixo Y secundário).
	Periodicidade da análise	**Mensal**
	Procedimento de Análise:	Identificar os projetos que não cumpriram a meta, procurar compreender os motivos do não cumprimento e planejar ações para que isto não se repita no mês seguinte. Uma análise complementar que deve ser realizada é a análise do histórico do indicador para verificar se este não cumprimento tem sido frequente ou se foi um caso especial.
	Responsável pela análise:	Líder de Time

Figura 1 – Exemplo de definição de indicador.

Indicador	Definição				Procedimento de Coleta, Armazenamento e Consistência					
	ID da Métrica	Métricas básicas	Descrição	Unidade	Responsável	Periodicidade	Coleta	Método	Armazenamento	Consistência
Entregas no Prazo	QTD_ENT_PRAZO	Quantidade de Entregas no Prazo	Quantidade de entregas no prazo	UND	Gerente de Projeto	Mensal	Consultar o relatório de acompanhamaneto do projeto, aba Marcos, e contabilizar as entregas realizadas no prazo no período corrente.	Manual	Armazene no DataDrill, na série e indicador referente a esta métrica, considerando o período correto da coleta.	Conversar com o SQE pra confirmar se o número de entregas planejadas está correto.
	QTD_ENT_PLAN	Quantidade de Entregas Planejadas	Quantidade total de entregas planejadas para o período	UND	Gerente de Projeto	Mensal	Consultar o relatório de acompanhamaneto do projeto, aba Marcos, e contabilizar as entregas planejadas para o período corrente.	Manual	Armazene no DataDrill, na série e indicador referente a esta métrica, considerando o período correto da coleta.	Conversar com o SQE pra confirmar se o número de entregas realizadas no prazo está correto.

Figura 2 – Exemplo de definição de indicador.

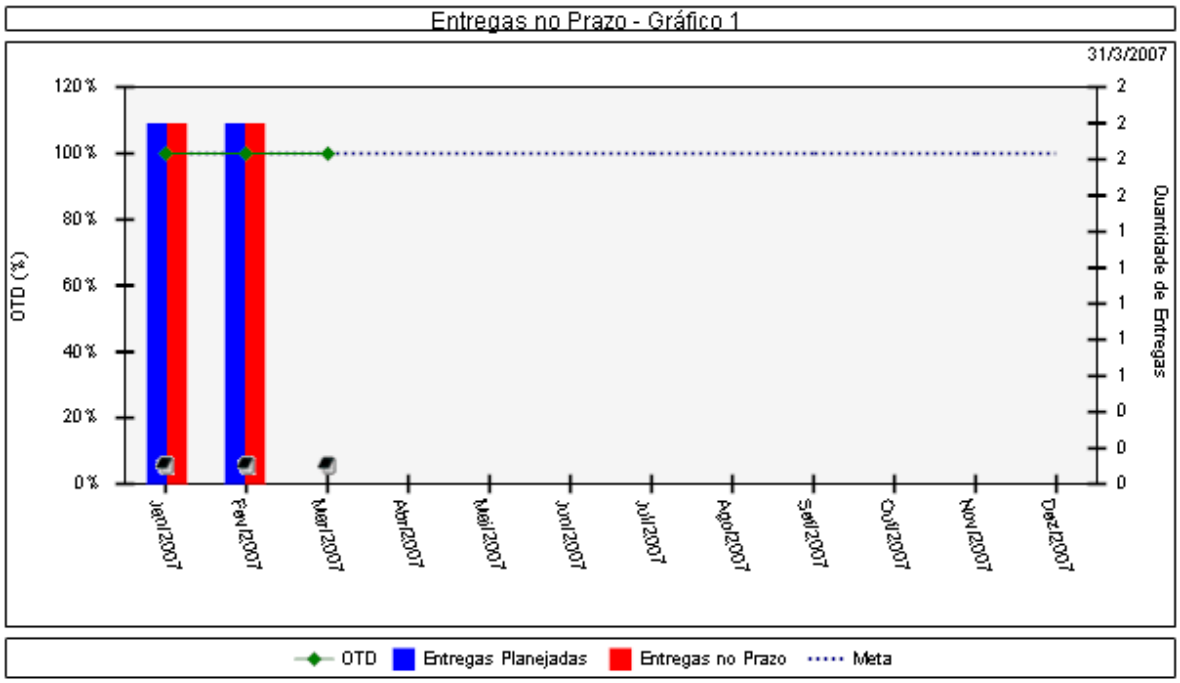

Figura 3 – Exemplo de apresentação de indicador.

Padrões relacionados:

- Padrão 1
- Padrão 2
- Padrão 3
- Padrão 4

Padrão 6: Não avaliar indivíduos

Contexto:

Em uma organização, muitos aspectos podem ser medidos, inclusive pessoas. Avaliar a performance dos indivíduos da organização tende a ser algo requerido naturalmente pela gerência, uma vez que é necessário ter conhecimento se as pessoas estão executando o que deveriam.

Problema:

- As pessoas (produtoras da informação) tendem a esconder a informação caso ela perceba que a tal informação pode gerar questionamentos sobre o desempenho do seu trabalho;
- O processo pode ser "burlado" para se obter um melhor valor da métrica, isto é, determinadas atividades serem executadas de forma direcionada para gerar um valor de métrica, mesmo que o objetivo final daquela atividade nem sempre seja atingido, porém a meta associada à métrica, sim;
- O benefício individual passa a ser mais importante do que o benefício do time ou organização. Muitas vezes, por medo de perder o emprego, o indivíduo passa a prezar apenas pelo alcance de suas metas para garantir seu lugar na organização, porém deixa de olhar para o sucesso da equipe da qual faz parte.

Solução:

- Deve ficar claro que o objetivo do programa de medição é:
  - Entender como o software está sendo construído;
  - Permitir que decisões podem ser tomadas com mais segurança;
  - Avaliar o impacto das mudanças.
- Definir níveis de privacidade de algumas informações, de maneira que determinadas informações de métricas sejam exclusivas para certos grupos, como o dos gestores, por exemplo.

Exemplo:

- O Efeito Hawthorne [13] [14] [15] descreve uma mudança no comportamento ou no desempenho de funcionários em resposta a mudanças ambientais. O nome é originário das experiências de cientistas na fábrica Hawthorne, que originou o desenvolvimento do campo das relações humanas na administração, em contraposição aos conceitos *tayloristas*.

  A proposta inicial era estudar os efeitos da iluminação na produtividade dos indivíduos. A hipótese é que o aumento dos níveis de luminosidade provocaria um aumento de produtividade dos indivíduos da fábrica. Os pesquisadores realizaram alguns estudos, em que:

  - **Estudo 1:** Produtividade melhorada em três departamentos, com a iluminação aumentando ou diminuindo.

  - **Estudo 2:** Dois grupos foram testados, em um deles não se melhorou a iluminação e no outro a iluminação foi melhorada a cada passo → Não houve diferença entre a produtividade dos grupos.

  - **Estudo 3:** Dois grupos foram testados, em um deles a iluminação manteve-se constante e no outro a iluminação foi diminuída a cada passo → Não houve diferença entre a produtividade dos grupos.

  - **Estudo 4:** Duas mulheres foram testadas → A produção das duas manteve-se constante, mesmo após grandes variações na iluminação.

  A conclusão a que se chegou a partir desse experimento é de que a produtividade aumenta quando há a percepção dos trabalhadores que a direção da empresa dá atenção a eles. Isto é, o fato psicológico é muito mais influente que o fisiológico.

  Dentro deste Padrão, pode-se dizer que o indivíduo pode levar as métricas a seguir um determinado caminho de acordo com sua percepção sobre a mesma.

Padrões relacionados:

- Padrão 1
- Padrão 4
- Padrão 5

Padrão 7: Usar métricas para entender

Contexto:

Utilizar métricas para motivação a fim de obter determinados resultados, seja: recompensando pessoas ou projetos baseado em suas performances com base em uma ou duas métricas, ou publicando gráficos que mostram os resultados desejados e indesejados.

Problema:

- Os participantes tendem a esconder os dados reais;
- Gerentes focam no valor do "número";
- O programa de medição propicia a imposição de poder (exigindo comportamentos desejados).

Solução:

- Deixar claro para todos que os dados das métricas são meramente informativos e não com intuito de punir ou medir pessoas;
- Utilizar melhoria de processo para obter o comportamento desejado. O programa de medição deve ser utilizado para avaliar se os resultados (objetivos) estão sendo atingidos.

Exemplo:

- A experiência de Milgram [16] foi uma experiência científica desenvolvida e realizada pelo psicólogo Stanley Milgram. O objetivo era entender de que forma os indivíduos observados tendem a obedecer às autoridades, mesmo que essas contradigam o bom-senso individual. A experiência pretendia inicialmente explicar crimes inumanos do tempo do Nazismo. A idéia era testar até que ponto um cidadão comum poderia infligir dor a outra pessoa simplesmente porque essa ordem lhe foi dada.

  Na experiência básica, planejada, duas pessoas chegam a um laboratório psicológico para participar de um estudo sobre memória e aprendizado. Um é chamado de "professor" e outro de "aluno". O experimentador explica que o estudo diz respeito aos efeitos da punição no aprendizado. O aluno é conduzido a uma sala, senta-se numa espécie de miniatura de cadeira elétrica; seus braços são imobilizados para impedir movimentos excessivos e um eletrodo é preso ao seu pulso. Ele é informado de que lhe serão lidas listas de pares de vocabulários simples e que, então, será testada a sua

capacidade de lembrar-se da segunda palavra de um par quando ouvir novamente a primeira palavra. Sempre que cometer um erro, receberá choques elétricos de intensidade crescente. Entretanto, o verdadeiro objeto dessa experiência é o professor.

O professor é um paciente realmente ingênuo que veio ao laboratório para essa experiência, em resposta a um anúncio colocado em um jornal da localidade solicitando voluntários para um estudo científico da memória. O aluno, ou vítima é, na realidade, um ator que não recebe nenhum choque. O objetivo da experiência é ver até que ponto uma pessoa prosseguirá numa situação concreta e mensurável em que lhe é ordenado infligir uma dor crescente numa vítima que protesta.

Como resultados, das 40 pessoas que participaram como "professor", 25 foram até o final na aplicação dos choques, contrariando as previsões de que os mesmos parariam antes.

Padrões relacionados:

- Padrão 1

Padrão 8: Incentivar, preparar e envolver as pessoas

Contexto:

O estabelecimento de um programa de medição envolve muitas pessoas, que precisam estar motivadas e preparadas para recebê-lo.

Problema:

- Os participantes não entendem o que é esperado deles;
- Existe muita oposição ao programa;
- Somente um grupo isolado é responsável pelo processo de medição, e só assim o mesmo funciona.

Solução:

- Realizar treinamentos e descrever o programa;
- Os gerentes devem:
    - Compartilhar os resultados com seus subordinados;
    - Reportar os benefícios de ter os dados e descrever como as informações ajudam os gerentes a tomar decisões.

- Medições não devem ser vistas como um processo adicional que apenas traz *overhead*, mas sim, devem estar embutidas nos outros processos.

Exemplo: Não possui.

Padrões relacionados:

- Padrão 1

Padrão 9: Focar na interpretação / Análise das métricas

Contexto:

No ambiente corporativo, as decisões precisam ser tomadas rapidamente.

Problema:

- Tomar decisões rápidas, sem analisar corretamente toda a sistemática do desenvolvimento, podendo causa, por exemplo, aumento no número de bugs nos testes de sistemas mesmo após o investimento em qualidade.

Solução:

- Analisar criteriosamente o dado, de forma a identificar tendências;
- Ter pessoas preparadas, do domínio do que está sendo medido para analisar os dados;
- Aplicar métodos específicos de análise de dados: BI, estatística.

Exemplo: Não possui.

Padrões relacionados:

- Padrão 1

Padrão 10: Centralizar os dados

Contexto:

Diversos atributos, em locais diferentes, em momentos distintos são mensurados.

Problema:

- Dados tendem a estar dispersos e só acessíveis por grupos específicos;
- Os dados se perdem com o passar do tempo;
- Gera-se um *overhead* de coleta e transformação dos dados.

Solução:

- Centralizar os dados coletados em um repositório de medições com os devidos controles de acesso.

Exemplo:

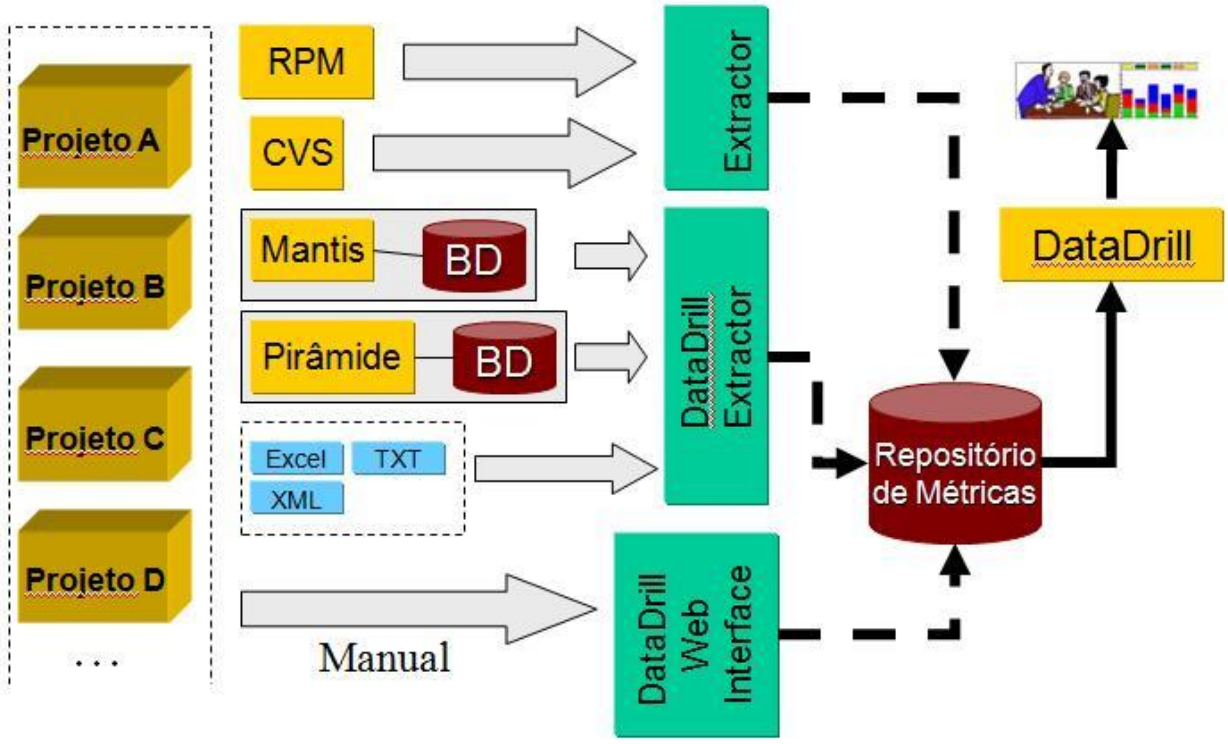

Figura 4 – Exemplo de como gerar um repositório de métricas para posterior recuperação.

Padrões relacionados:

- Padrão 1

5. Conclusões

Os padrões documentam uma parte repetida de um projeto, permitindo seu entendimento e aplicação em um contexto particular. Os padrões fornecem ao projeto um vocabulário comum aos *stakeholders*, facilitando a comunicação entre os mesmos. Eles acabam constituindo uma base de experiência para reuso de software (não apenas no nível técnico) através da exposição do conhecimento de vários especialistas e de alguém que documentou esse conhecimento observado ao longo dos anos.

É com esse intuito que se deseja criar uma base de conhecimento ligada aos padrões para implantação de programas de medição das organizações de software, obstáculo enfrentado pelas empresas que desejam institucionalizar um programa eficaz para medir a produtividade e qualidade dos produtos e serviços prestados pelas empresas.

6. Trabalhos futuros

Como proposta de trabalho futuro, deseja-se indetificar outros padrões relacionados à implantação de programas de medição, além de constatar sua aplicabilidade em um grupo maior de empresas a fim de tornar esse conjunto de práticas em padrões a serem utilizados pelos profissionais ligados ao desenvolvimento de software.

Também se tem como objetivo a correta categorização do conjunto de padrões definidos, como coleção, catálogo, sistema ou linguagem de padrões. Além disso, verificar se a estrutura proposta neste trabalho realmente se mostra como a mais adequada para o padrão proposto.

7. Bibliografia

[1] Software Reuse and Software Product Lines. Disponível em: http://www.biglever.com/technotes/reuse_spl.html?source=reuse. Acessado em: 11/01/2009.

[2] Wikipedia – Smalltalk. Disponível em: http://pt.wikipedia.org/wiki/Smalltalk. Acessado em: 11/01/2009.

[3] Visão geral sobre Padrões de Software (Software Patterns). Disponível em: http://knol.google.com/k/misael-santos/padres-de-software/irtziatj2kf9/2#. Acessado em: 11/01/2009.

[4] ALEXANDER, C. The Timeless Way of Building. Oxford University Press, New York, NY, 1979.

[5] ALEXANDER, C. et al. A Pattern Language: Towns, Buildings, Construction. Oxford University Press, New York, NY, 1977.

[6] ANDRADE, R.; MARINHO, F. Diagnóstico do reuso de padrões de software no Instituto Atlântico – Projeto: Métodos e Técnicas associados a Padrões de Software. Relatório Técnico do Projeto de Pesquisa e Desenvolvimento em Colaboração Departamento de Computação / Universidade Federal do Ceará e Instituto Atlântico (Convênio CVE/P&D/005/02), Fortaleza, CE, Fev. 2003.

[7] Padrões de Projeto de Sofware. Disponível em: http://pt.wikipedia.org/wiki/Padr%C3%B5es_de_projeto_de_software. Acessado em: 11/01/2009.

[8] ANDRADE, R.; MARINHO, F. Diagnóstico do reuso de padrões de software no Instituto Atlântico – Projeto: Métodos e Técnicas associados a Padrões de Software. Relatório Técnico do Projeto de Pesquisa e Desenvolvimento em Colaboração Departamento de Computação / Universidade Federal do Ceará e Instituto Atlântico (Convênio CVE/P&D/005/02), Fortaleza, CE, Fev. 2003.

[9] Notas Didáticas – Sugar Loaf PLoP 2005. Disponível em: http://sugarloafplop2005.icmc.usp.br/NotasDidaticasPadroes.pdf. Acessado em: 11/01/2009.

[10] Appleton, Brad. Patterns and Software: Essential Concepts and Terminology. Disponível em: http://www.enteract.com/~bradappdocpatterns-intro.html.

[11] Buschmann, F. et at. A System of Patterns, Wiley, 1996.

[12] James O. Coplien. Software Design Patterns: Common Questions and Answers. In Linda Rising, editor, The Patterns Handbook: Techniques, Strategies, and Applications, p. 311-320. Cambridge University Press, New York, January 1998.

[13] Site Finanças Comportamentais. Disponível em: http://financascomportamentais.blogspot.com/2008/02/efeito-hawthorne.html. Acessado em 14/01/2009.

[14] Wikipédia – Experiência de Hawthorne (português). Disponível em: http://pt.wikipedia.org/wiki/Experi%C3%AAncia_de_Hawthorne. Acessado em: 14/01/2009.

[15] Wikipédia – Hawthorne Effect (inglês). Disponível em: http://en.wikipedia.org/wiki/Hawthorne_effect. Acessado em: 14/01/2009.

[16] Milgran, S. "Os perigos da obediência". Behavioral Study of Obedience, *Journal of Abnormal and Social Psychology, 67, 371-378. Tradução do Consulado dos Estados Unidos no Rio de Janeiro – Revista Diálogo*. Disponível em: http://www.bernardojablonski.com/pdfs/graduacao/perigos_obediencia.pdf. Acessado em 01/01/2009.

[17] Aquino, G. *Padrões para implantação de programas de medição em organizações de software*. SugarLoafPLoP-2007.