

1 Introdução

clairvoyant (klâr-'voi-&nt): *adj.* Unusually perceptive
Merriam Webster's Dictionary

1.1. Motivação e contexto

1.1.1. A importância da medição e análise para a engenharia de software

Ambientes de desenvolvimento e engenharia de software variam muito em sua forma, mas costumam ter os mesmos objetivos básicos: a produção de software de qualidade com o menor custo e no menor tempo possível. Para o funcionamento ótimo destes ambientes, as decisões tomadas precisam ser as mais acertadas possíveis. Assim sendo, faz-se necessário tomar essas decisões baseadas em fatos e não em suposições. Isso nos leva à necessidade de fazer medições acerca do ambiente em questão e analisá-las. Não é surpreendente, então, constatarmos que várias referências respeitadas de engenharia de software (SEI, 2002; Humphrey, 1995; SOFTEX, 2006) consideram a medição e análise de software disciplinas fundamentais ao bom funcionamento dos ambientes de desenvolvimento.

Os dados devidamente analisados poderiam ser utilizados como insumo para tomada de decisões no ambiente. Os benefícios trazidos pela prática da medição e análise são principalmente de dois tipos: gerenciamento de projeto e gerenciamento de processo.

Entre as disciplinas de gerência de projeto mais afetadas, podemos citar:

- planejamento de projeto, por meio de estimativas de custo, prazo e equipe baseadas em analogias com dados históricos
- controle de projeto, com acompanhamento de prazo, custo, esforço e qualidade baseado em medições do projeto em andamento

- gestão de risco, já que adquirimos a possibilidade de calcular a probabilidade de um dado problema se tivermos os dados referentes a esse problema

Apesar dos objetivos principais do desenvolvimento de software estarem ligados a atributos de produto e projeto, alcançar os níveis desejados de tais atributos se torna mais fácil num ambiente onde o processo de desenvolvimento é medido e controlado. Por exemplo, a eficácia de remoção de defeitos poderia ser medida num processo com esse fim, nos dando visibilidade sobre o funcionamento desse processo.

A análise de desempenho de processos possibilitada pela medição e análise pode ser altamente benéfica, pois a melhoria de um processo de desempenho deficiente traz um impacto positivo aos projetos conseguintes. Se os processos do ambiente forem controlados, torna-se possível implantar métodos de melhoria contínua de processos conhecidos. Conseqüentemente, intervalos de controle para desempenho de processo podem ser estabelecidos, assim possibilitando a determinação de valores-limite para a tomada de ações corretivas. Desta maneira, cresce a nossa capacidade de monitoramento do processo.

Quanto ao gerenciamento de processo, as principais influências positivas da medição e análise se encontram nas seguintes disciplinas:

- controle de processo, através do acompanhamento de indicadores-chave do processo
- avaliação de efetividade de iniciativas de melhoria contínua, valendo-se da análise de indicadores pré e pós-iniciativa

1.1.2.

O repositório como solução para problemas de medição

Levando-se em conta a utilidade evidente da implantação das disciplinas de medição e análise num ambiente de engenharia de software, a taxa de adoção destas práticas é surpreendentemente baixa. Isso ocorre porque elas são difíceis de serem executadas, apresentando alguns obstáculos inerentes.

Os obstáculos de medição e análise podem ser de vários tipos, como falta de alinhamento com os objetivos institucionais, mau planejamento da iniciativa de medição, custo de medição, entre outros. As principais contribuições ligadas ao

uso de um repositório de medições estão ligadas à diminuição dos seguintes esforços:

- **Esforço de integração:** as ferramentas geradoras de dados de medição (e.g., contadores de linhas de código) não se integram diretamente com as ferramentas de análise (como planilhas eletrônicas ou aplicativos de análise estatística avançada).
- **Esforço de registro:** geralmente as medições coletadas necessitam passar por transformações para serem armazenadas conforme o modelo conceitual do repositório.

As capacidades de armazenamento e resgate centralizadas de um repositório de medições de software podem ajudar a contornar esses problemas. Por exemplo, o esforço de integração pode ser reduzido se o repositório centralizar as informações de medição: todos os dados coletados sendo registrados no repositório e sendo obtidos para análise a partir do repositório. Já o esforço de registro pode ser reduzido se o repositório oferecer uma interface de importação de dados flexível, com capacidade de descrever e executar as transformações necessárias para o registro dos dados.

Além das questões levantadas, um repositório de medições traz outros benefícios. Um repositório unificado é necessário para combinar diferentes quantificações, pois não existe uma única ferramenta de medição abrangendo todas as necessidades de uma organização desenvolvedora de software. Ele permite análise, avaliação e comparação de diferentes aspectos medidos na área de desenvolvimento de software (Foltin e Dumke, 1998). Por fim, a existência de um repositório organizacional de medições é uma demanda explícita em alguns processos de melhoria contínua, por exemplo, (SEI, 2002).

1.1.3.

A evolutibilidade como requisito de repositórios de medição

(Harrison, 2004) aponta, entre outras, duas limitações de repositórios de medição contemporâneos: a incapacidade de lidar com obsolescência de medições e o aumento do conjunto de medições, ou seja, com a descontinuidade da coleta de algumas medições e o surgimento de outras novas para serem coletadas. A superação dessas limitações por um repositório de medição é importante, pois o

ciclo de coleta e análise de medições é evolutivo: quanto mais sabemos de nosso ambiente de engenharia de software, melhor sabemos de quais informações precisamos.

No escopo deste trabalho, vamos chamar de *evolutibilidade*, ou seja, a capacidade de *evolução* de um repositório de medição, a capacidade de mudança no seu modelo de dados durante a sua operação. Isso abrange a mudança no conjunto de medições colhidas e a mudança no conjunto de possíveis relacionamentos entre entidades sobre as quais se colhem métricas.

1.1.4.

A complexidade associada à evolutibilidade

A evolução do modelo de medição do repositório seria relativamente simples se supuséssemos que o sistema gestor de bancos de dados subjacente ao repositório possui funcionalidades de evolução de esquema. Como um percentual ínfimo da base instalada de sistemas gestores de bancos de dados obedece a essa restrição, partiremos do pressuposto que o repositório opera sobre um SGBD desprovido de tais capacidades.

Existem algumas maneiras para dotar um repositório de medição de capacidade de evolução (Franca et al., 1999) (Harrison, 2004). Entretanto, elas podem acabar expondo o usuário a formas demasiadamente complexas de associação entre medições e entes medidos. Desta maneira, a complexidade das operações de definição, inserção e consulta aos dados do repositório prejudicaria a usabilidade desse repositório. Em compensação, um repositório dotado de capacidade evolutiva sem tal prejuízo à sua usabilidade seria de elevada utilidade à disciplina de medição de software.

1.2.

Problema proposto

Tendo em vista as considerações sobre evolução, propomos, no âmbito deste trabalho, apresentar uma abordagem para dispor de evolução transparente num repositório de medição genérico, sem necessitar contar com funcionalidades de evolução de esquema no SGBD subjacente.

Quando mencionamos evolução *transparente*, queremos dizer que a evolutibilidade do repositório é considerada característica intrínseca ao repositório. Dito de outra maneira, o repositório deverá resolver internamente problemas associados à capacidade evolutiva, evitando que os seus usuários fiquem expostos a esses problemas. Assim sendo, o usuário não precisará entender como funciona o mecanismo de evolução do repositório para utilizá-lo. Queremos também conter o impacto de alterações no modelo de medição, para que usuários do repositório não sejam impactados por mudanças no modelo de dados de medição feitos por outros usuários.

Já ao demandar que o repositório seja genérico, queremos dizer que ele não imporá restrições a:

- O tipo de métricas coletadas (esforço, tamanho físico, tamanho funcional, etc.)
- O ferramental utilizado no apoio ao desenvolvimento e à manutenção
- A linguagem de programação (e, por conseguinte, o paradigma de desenvolvimento)
- O processo de desenvolvimento de software utilizado
- O ferramental utilizado para analisar os dados de medição

1.3. Abordagem

A abordagem para resolver o problema proposto passa pela concepção de um meta-modelo de medição, um modelo de consultas em repositórios de medição baseado nesse meta-modelo e a construção de um repositório de medição baseado nesses modelos, permitindo a validação da abordagem apresentada para a evolução transparente (e conseqüentemente dos modelos) por meio de uma prova de conceito.

A concepção do meta-modelo de dados de medição passa pelo estudo de meta-modelos de dados publicados na literatura, e pela seleção de características baseada nos benefícios oferecidos por cada modelo. Já o modelo de consultas foi inspirado na álgebra relacional estendida (Silberschatz et al., 1999), com algumas

extensões visando a facilitar os tipos de análises costumeiras para medições de software.

O sistema *Clairvoyant*, baseado nos modelos de dados e consulta, foi construído para ter algumas funções básicas de repositório como, por exemplo, importação de dados. São estudados alguns exemplos de repositórios de medição, do ponto de vista de funcionalidade oferecida, e desse conjunto foi escolhido um subconjunto considerado essencial. No mais, seu conjunto de funcionalidades deve ser suficiente para permitir uma prova de conceito de sua abordagem.

Finalmente, há a validação do sistema por meio de uma prova de conceito. Essa prova consistirá em alterar o esquema de medições, adicionando e removendo atributos de entidades; e efetuar consultas sobre o repositório, verificando se o resultado dessas consultas confere com o resultado esperado.

1.4. Contribuição

As contribuições aportadas por este trabalho se dividem em dois tipos: teóricas (requisitos de evolução, meta-modelo de medição, modelo de consulta) e práticas (o sistema *Clairvoyant*).

Uma primeira contribuição deste trabalho é a própria discussão de requisitos de evolução em repositórios de medição, pois ela é quase inexistente em outros trabalhos e muito importante para derivarmos os requisitos do sistema *Clairvoyant*.

Outra contribuição importante é o modelo de dados (meta-modelo de medição) proposto: ele apresenta faz parte de uma solução para o problema da evolução em repositórios de medição genéricos.

Também devemos ressaltar o modelo de consulta, pois ele permite a transparência da evolução em repositórios com o modelo de dados proposto. Em menor grau, a própria discussão sobre modelos de consulta em repositórios de medição é uma contribuição. É um assunto pouquíssimo discutido na literatura e que pode ajudar a resolver vários outros problemas em medição de software como, por exemplo, facilitar a análise de dados de medição.

Como última contribuição, podemos citar o sistema *Clairvoyant*, que serve como prova de conceito para esses modelos. Ele foi projetado para ser facilmente

expansível, de modo a poder resolver o problema proposto num contexto mais amplo.

1.5. Estrutura do trabalho

Neste capítulo, o trabalho é apresentado em suas linhas gerais: qual problema ele deseja solucionar e seu domínio, a solução obtida para o problema, qual é a abordagem proposta para resolver se chegar à solução e qual a contribuição deste trabalho.

No capítulo 2, será exposta a abordagem deste trabalho diante das disciplinas de medição e análise de software, evidenciando com quais problemas ele pretende auxiliar. Pretendemos expor conceitos e problemas com repositórios de medições de software, Finalmente, pretendemos enumerar alguns de seus problemas e expor soluções para eles, que serão desenvolvidas em capítulos posteriores.

No capítulo 3, entraremos no detalhe de como o sistema *Clairvoyant* resolve os problemas que constam no seu escopo. Será dada uma explicação detalhada de sua abordagem para a evolução transparente em repositórios de medição. Além disso, será explicado como usar o sistema para resolver os problemas propostos.

No capítulo 4, faremos um estudo da literatura relacionada a este trabalho. Pretendemos tanto evidenciar quais trabalhos serviram de base para este, como também apresentar outros trabalhos com abordagens diferentes para atingir objetivos relacionados aos deste trabalho.

No capítulo 5, serão relatados os procedimentos de validação aos quais o sistema *Clairvoyant* se sujeitou no âmbito deste trabalho e quais foram os resultados conseqüentes. Será feito um comentário a respeito da efetiva capacidade do sistema de contribuir com a resolução dos problemas propostos.

No capítulo 6, concluiremos relatando as lições aprendidas ao desenvolver este trabalho e apontaremos direções futuras para trabalhos que queiram prosseguir nos objetivos deste.