Um estudo sobre métricas de produto e vulnerabilidades para tomada de decisões

Arthur Del Esposte Carlos Bezerra Paulo Meirelles Hilmer Neri
Universidade de Brasília - Faculdade UnB Gama, Brasil

UnB Gama
O novo endereço da tecnologia

Resumo

Neste trabalho é explorada a importância da utilização de métricas estáticas de código-fonte para suportar a tomada de decisões, tanto a nível técnico quanto gerencial a respeito do design e segurança do software. Além disso, é proposta uma nova técnica para realizar medições que será viabilizada a partir da evolução de duas ferramentas de monitoramento de código-fonte.

Design, Segurança e Métricas de Software

- Qualidade interna é um dos principais fatores de sucesso de projetos de software
  - Mais testes automatizados
  - Legibilidade e compreensão
  - Reduzem riscos de inserção de *bugs*
  - Aumentam as oportunidades de encontrar e tratar vulnerabilidades
- 80% das vulnerabilidades exploráveis estão ligadas à má codificação segundo o estudo ICAT/NIST (2005)
- A qualidade interna do software está, portanto, relacionado a sua segurança. A aplicação de princípios de segurança podem envolver a aplicação de princípios de *design*. A aplicação de práticas e técnicas de *design* são fundamentais para o desenvolvimento de códigos seguros.
- A medição pode ser utilizada como ferramenta para apoiar o acompanhamento e tomada de decisões sobre a segurança e qualidade interna, através de indicadores e metas que indiquem oportunidades de melhorias.

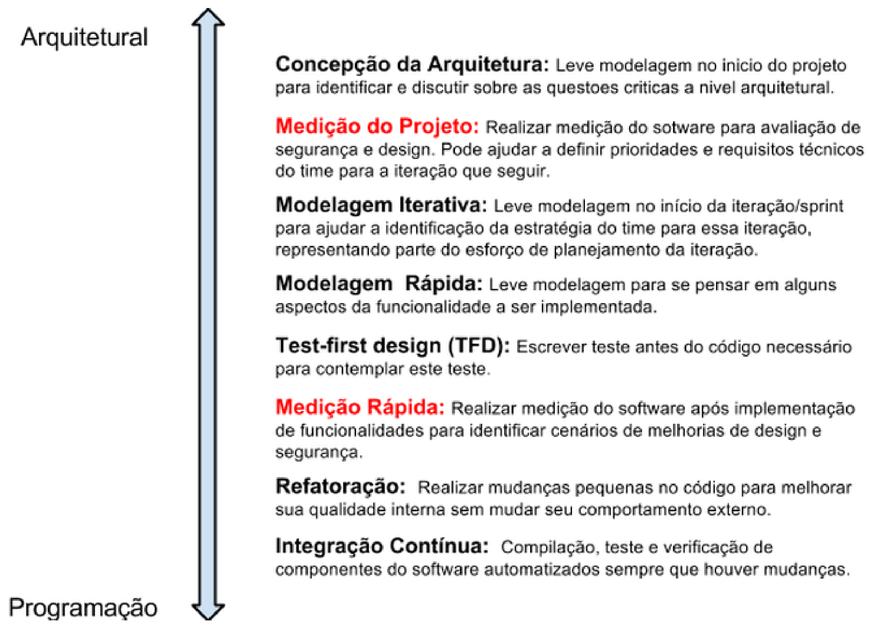

Figura : Proposta de uso de métricas como práticas ágeis

- Dificuldades: grande quantidade de métricas, coletas manuais, poucos recursos de visualização, dificuldades de compreensão de valores, interpretações errôneas e uso de métricas isoladas.

Cenários de Decisão

- Os Cenários de Decisão visam nomear e mapear estados observáveis através de métricas de código-fonte que indicam a existência de determinada característica dentro do software, classe ou método, potencializando o uso de métricas para tomada de decisões em projetos. A estrutura dos cenários consistem em:
  - **Nome**: Identificação única do cenário
  - **Métricas Envolvidas**: Métricas necessárias para a caracterização do cenário
  - **Nível**: Abstração envolvida (projeto, classe, método)
  - **Descrição**: Discuti os problemas, princípios envolvidos e a caracterização
  - **Caracterização com Métricas**: Define e discuti como as métricas envolvidas devem ser utilizadas para identificar o cenário
  - **Ações Sugeridas**: Propõe um conjunto de ações específicas tais como uma refatoração, a utilização de um padrão de projeto, prática e aplicação de princípios
- Projetos podem utilizar cenários de referência ou até mesmo definir novos cenários de acordo com parâmetros de qualidade do projeto.

Formação de Cenários de Design e Segurança

- Verificou-se uma forte relação entre princípios de design de software com os princípios de design seguro, onde a aplicação de ambos podem prover softwares mais robustos, extensíveis e seguros.
- Vulnerabilidades específicas são difíceis de se encontrar, exigindo o conhecimento do profissional a respeito dessa vulnerabilidade. Porém, mesmo com tal conhecimento, a identificação dessas vulnerabilidades em um código com "mau"design torna-se uma tarefa muito difícil.
- Os cuidados com o bom design do código-fonte são fundamentais para o desenvolvimento de códigos seguros, podendo ser realizados através, por exemplo, da prática de refactorings e aplicação de padrões de projeto e não somente através do tratamento de vulnerabilidades específicas.
- A complexidade, por exemplo, é uma característica de design que está diretamente relacionada com a segurança. É um dos principais problemas que afetam a qualidade interna do software, dificultando principalmente a manutenção e evolução do software, aumentando riscos de inserção de bugs e vulnerabilidades e dificultando o tratamento dos mesmos. Características que podem indicar complexidade:
  - Grande número de métodos, de filhos e de métodos públicos de uma classe
  - Árvores de herança profundas
  - Alto acoplamento
  - Baixa Coesão
- A utilização de métricas de código-fonte e de vulnerabilidades podem auxiliar o desenvolvimento de softwares mais robustos.
- A partir do estudo de relação entre design e segurança, foi proposto um conjunto de Cenários de Decisão voltados principalmente para caracterizar problemas de segurança em projetos de software.

Cenários de Decisões no Contexto de Segurança de Software

Cenário	Caracterização com Métricas	Ações Sugeridas
Alta Superfície de Ataque a Atributos Internos	Valor alto de NPA/NOA	**Refatorações**: Encapsulate Field; **Aplicar Princípios**:Redução da superfície de ataque, Princípio de encapsulamento
Alta Superfície de Ataque Operacional	Valor alto de número NPM e MNPM	**Refatorações**: Hide Method; Remove Parameter; **Aplicar Princípios**:Princípio de Encapsulamento; **Aplicar padões**: Padrão facade
Ponto Crítico de Falha	Alto valor de ACC+NOC	**Refatorações**: Extract Class; **Aplicar Princípios**:Princípios de Distribuição de Responsabilidades GRASP
Confidencialidade Ameaçada	Valor acima do aceitavel de OBAA + DUPV	garantir o range de acesso de um array, realizando verificações do índice de acesso antes de acessar o valor da variável.
Uso de Variáveis não inicializadas	Valor acima do aceitavel de AUV + UAV + ROGU	Inicializar variáveis no momento que foram instanciadas, ou garantir que a variavel será preenchida antes de ser usada.

Tabela : Parte II - Resumo de todos os cenários

Trabalhos Futuros

- Implementação de Cenários de Decisões na plataforma livre de monitoramento de código fonte Mezuro e em um ambiente de Data Warehousing.
- Criar configurações nas duas ferramentas para viabilizar a utilização dos cenários de segurança propostos.
- Estudo de correlação estatística entre métricas de design e de segurança a partir da análise de softwares livres para apoiar cientificamente a concepção de novos cenários de referência.
- Propor novos Cenários de Decisão.