Modelo de processo para desenvolvimento de aplicações seguras

Tarcizio Vieira Neto
OWASP member
SERPRO
tarcizio.vieira@owasp.org

OWASP
AppSec LATAM 2011
06/10/2011

Copyright © The OWASP Foundation
Permission is granted to copy, distribute and/or modify this document under the terms of the OWASP License.

The OWASP Foundation
http://www.owasp.org

Objetivos da Apresentação

- Apresentar uma visão geral dos processos de segurança definidos pela OWASP (CLASP e OpenSAMM)

- Apresentar uma sugestão de processo em linguagem BPMN

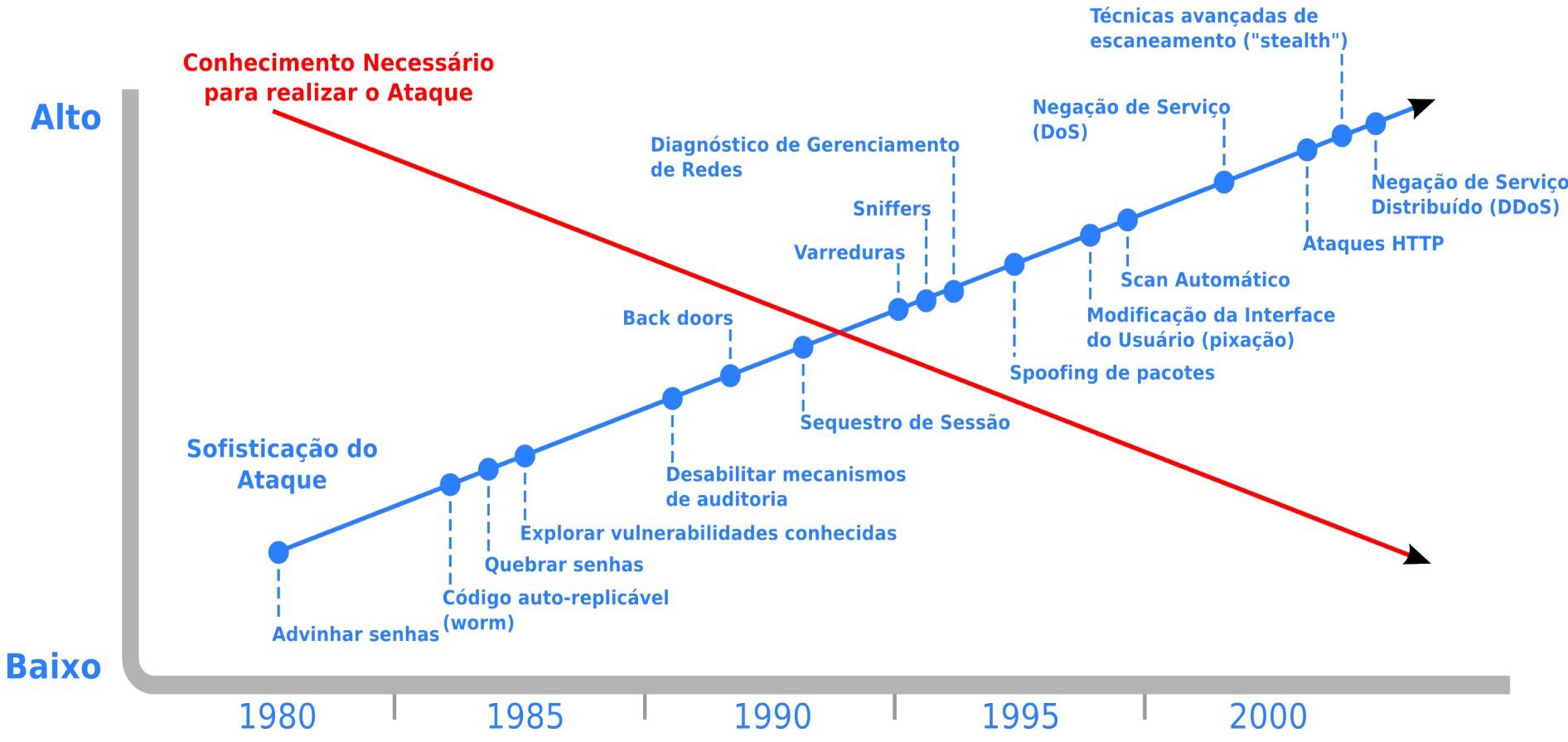
Roteiro

- Introdução
- Especificações da OWASP
- Desenho do Processo
- Conclusões
- Referências

Introdução

- Problemática atual
  - Segundo o SANS: "Nos últimos anos, a quantidade de vulnerabilidades descobertas em aplicações é muito maior que as descobertas em sistemas operacionais."
  - Segundo o Gartner Group, 75% dos ataques acontecem na camada de aplicação.
  - Segundo o NIST, 92% das vulnerabilidades estão no software.

Sofisticação dos Ataques x Conhecimento Necessário



Introdução

- Problemática atual
  - Segurança de aplicações não está restrita apenas no processo de codificação
  - Para desenvolver software de forma segura, é necessário **pensar em segurança nas demais fases do ciclo de desenvolvimento**, envolvendo:
    - Programadores
    - Arquitetos de software
    - Engenheiros de requisitos
    - Analista de Negócio

Introdução

- Problemática atual
  - MITOS:
    - **"O desenvolvedor irá lidar com questões de segurança"**
      - Deve ser solicitado e formalmente acordado.

Introdução

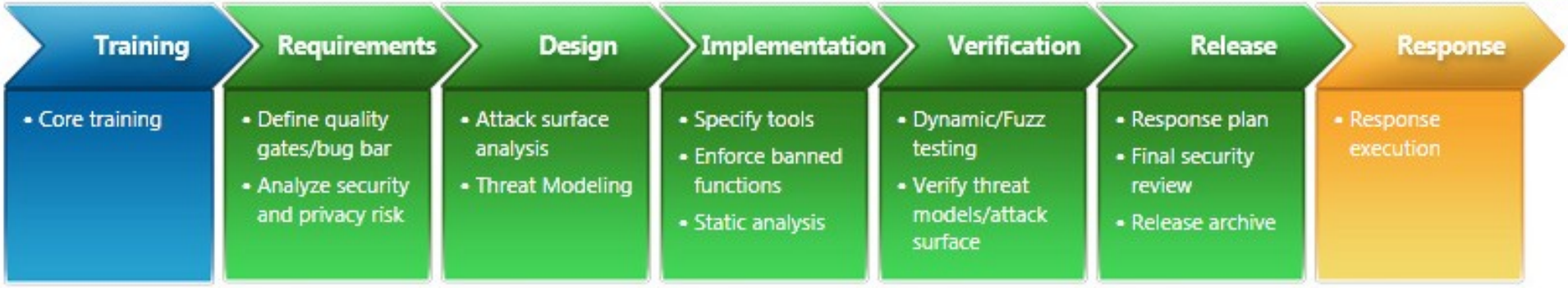
- Problemática atual
  - MITOS:
    - **"Vamos investir em treinamentos para os desenvolvedores e/ou contratar desenvolvedores com conhecimentos em segurança..."**
      - E os analistas de negócio e arquitetos de software?

Introdução

- Problemática atual
  - MITOS:
    - **"Adquirimos o melhor firewall de aplicação do mercado,... logo estamos seguros"**
      - Firewalls de aplicação não conseguem resolver todos os problemas de segurança (principalmente falhas nas funções de negócio)

Introdução

- Ciclo de desenvolvimento seguro
  - Microsoft SDL



Introdução

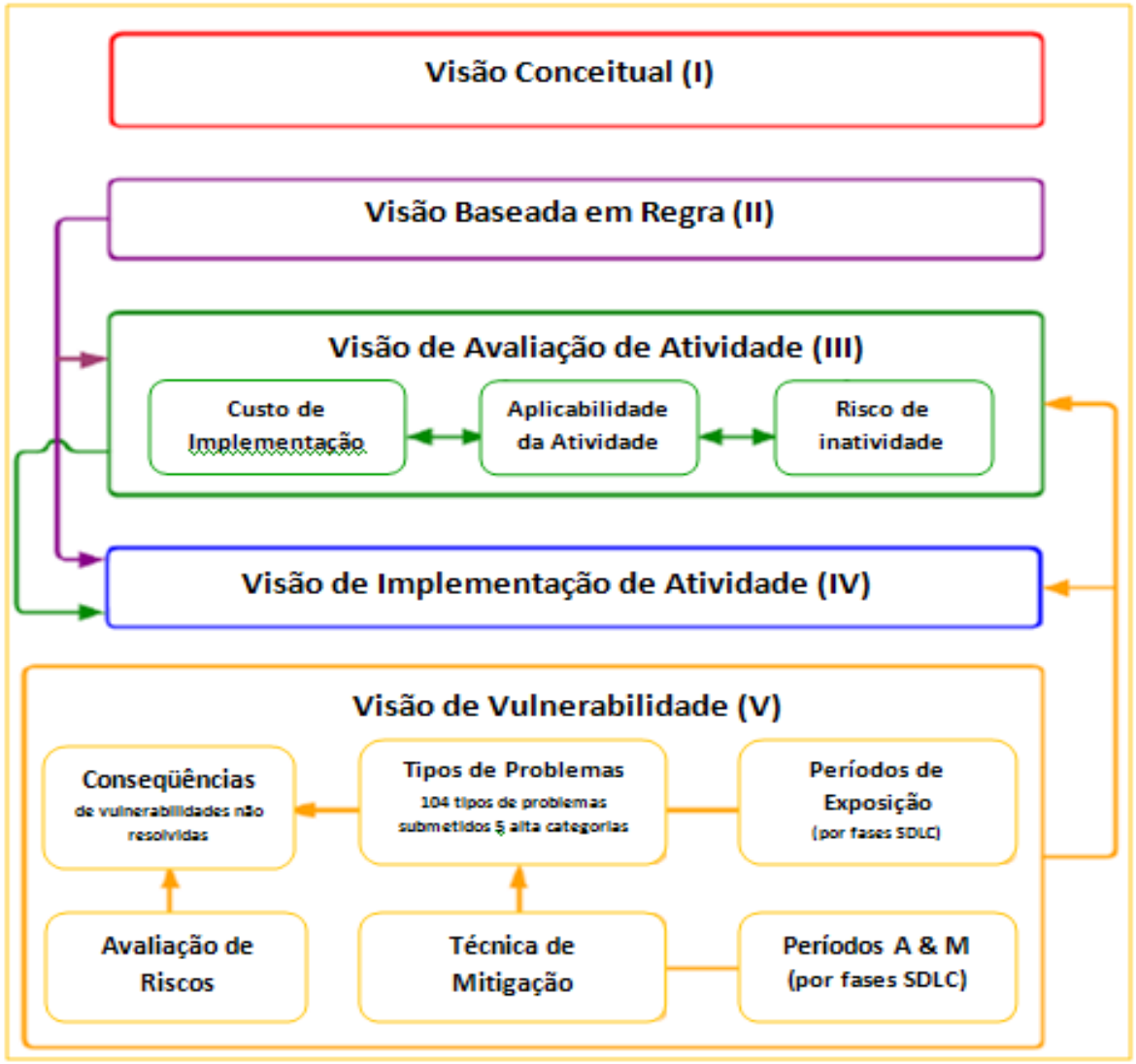
- Ciclo de desenvolvimento seguro – SDL
  - Requisitos
  - Projeto / Design
  - Implementação (Codificação Segura)
  - Verificação
  - Release (liberação de versões)
  - Resposta

Especificações da OWASP

- Secure Software Development Process (OWASP CLASP)
  - CLASP - Comprehensive, Lightweight Application Security Process
  - 7 Boas Práticas em Segurança de Aplicações
  - Cobre todo o ciclo de desenvolvimento de software
- Adaptável a qualquer processo de desenvolvimento de software
  - Define papéis no SDLC
  - 24 componentes de processos baseados em papéis

Especificações da OWASP

- Visões CLASP



Especificações da OWASP

- Visões CLASP
  - **Visão Conceitual (I)** apresenta uma visão geral de como funciona o processo CLASP e como seus componentes interagem. São introduzidas as melhores práticas, a interação entre o CLASP e as políticas de segurança, alguns conceitos de segurança e os componentes do processo.

Especificações da OWASP

- Visões CLASP
  - **Visão baseada em Regras (II)** introduz as responsabilidades básicas de cada membro do projeto (gerente, arquiteto, especificador de requisitos, projetista, implementador, analista de testes e auditor de segurança) relacionando-os com as atividades propostas, assim como a especificação de quais são os requisitos básicos de cada função.

Especificações da OWASP

- Visões CLASP
  - **Visão de Avaliação de Atividades (III)** descreve:
    - **propósito** de cada atividade
    - **custo** de implementação
    - **aplicabilidade**
    - **impacto** relativo aos **riscos** em caso de não se aplicar a atividade.

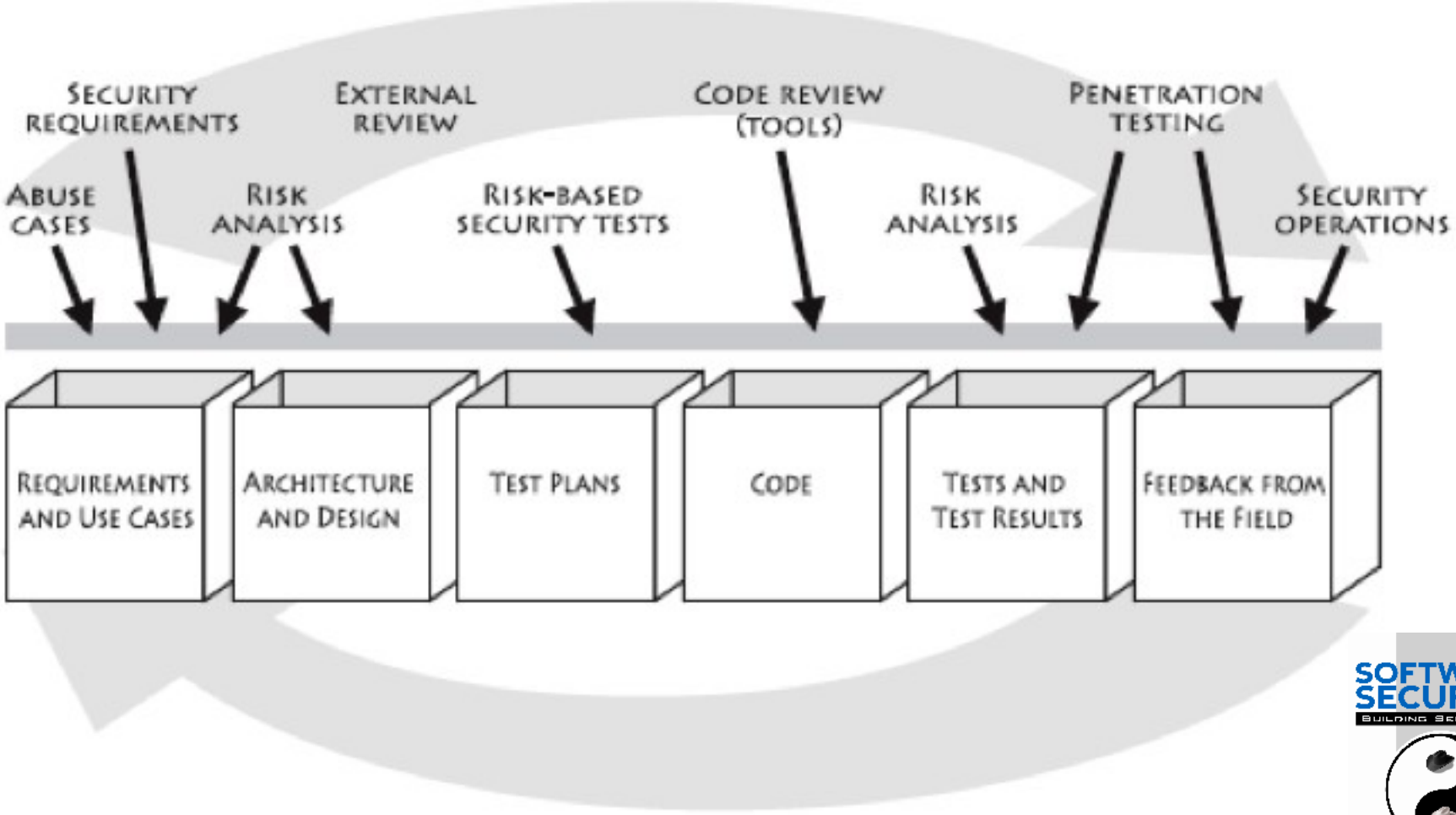
Especificações da OWASP

- Visões CLASP
  - **Visão de Implementação (IV)** descreve o conteúdo das 24 atividades de segurança definidas pelo CLASP e identifica os responsáveis pela implementação, bem como as atividades relacionadas.

Especificações da OWASP

- Visões CLASP
  - **Visão de Vulnerabilidades (V)** possui um catálogo que descreve 104 tipos de vulnerabilidades no desenvolvimento de software, divididas em cinco categorias:
    - Erros de Tipo e Limites de Tamanho;
    - Problemas do Ambiente;
    - Erros de Sincronização e Temporização;
    - Erros de Protocolo e Erros Lógicos em Geral.
  - Nessa atividade também é realizada técnicas de mitigação e avaliação de risco. Assim como período de A & M (Avoidance e Mitigation) por fase do SDLC.

Gary McGraw's and Cigital's model



SOFTWARE SECURITY
BUILDING SECURITY IN
GARY McGRAW

Visão Geral

Microsoft SDL – Completo (pesado),bom para grandes empresas

Modelo de McGraw e Cigital – Alto nível, com poucos detalhes

CLASP – Grande coleção de atividades, mas sem prioridades definidas

Todos são bons modelos para serem utilizados por especialistas na forma de guia, mas se tornam difíceis para pessoal que não trabalha na área de segurança.

Especificações da OWASP

- Software Assurance Maturity Model (SAMM)

Motivações:

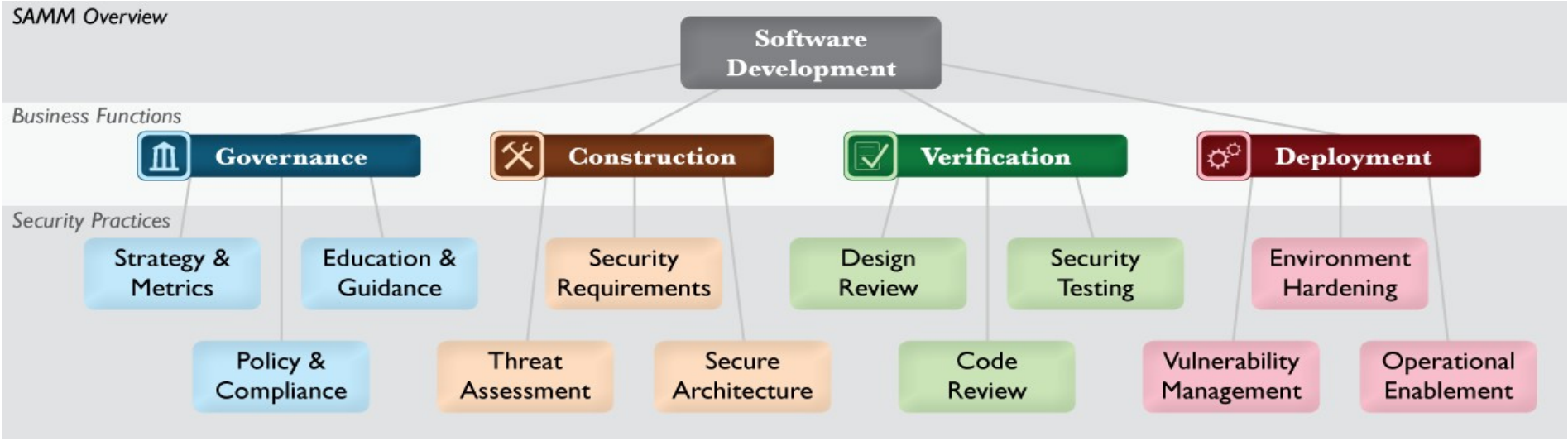
- O comportamento de uma organização muda muito lentamente
- As mudanças devem ser iterativas enquanto se trabalha para atingir objetivos de longo prazo
- Não existe uma "receita de bolo" que funciona para todas as organizações
- Uma solução precisa permitir abordagem baseada em riscos para se tomar decisões
- Uma solução precisa prover detalhes suficientes para equipes que não lidam diariamente com segurança
- De modo geral os processos devem ser simples, bem definidos e mensuráveis

Especificações da OWASP

- Software Assurance Maturity Model (SAMM)
  - Subdividido em funções de negócio:

Especificações da OWASP

- Software Assurance Maturity Model (SAMM)
  - 3 Práticas de segurança para cada função de negócio
  - As práticas de segurança cobrem todas as áreas relevantes para a garantia de segurança de software



Especificações da OWASP

- Software Assurance Maturity Model (SAMM)
  - Cada prática possui 3 objetivos que especificam como podem ser melhorados
  - Estabelece a noção de níveis de práticas a serem alcançadas
  - Os três níveis de cada prática geralmente correspondem a:
    - (0: Ponto de partida implícito, sem prática implementada)
    - 1: Entendimento inicial da Prática e implementação ad hoc
    - 2: Aumento da eficiência e/ou efetividade da Prática
    - 3: Domínio completo da Prática em escala

Education & Guidance

...more on page 42

	EG 1	EG 2	EG 3
OBJECTIVE	**Offer development staff access to resources around the topics of secure programming and deployment**	**Educate all personnel in the software life-cycle with role-specific guidance on secure development**	**Mandate comprehensive security training and certify personnel for baseline knowledge**
ACTIVITIES	A. Conduct technical security awareness training B. Build and maintain technical guidelines	A. Conduct role-specific application security training B. Utilize security coaches to enhance project teams	A. Create formal application security support portal B. Establish role-based examination/certification

Especificações da OWASP

- Para cada nível o SAMM define:
  - Objetivos
  - Atividades
  - Resultados
  - Métricas de Sucesso
  - Custos
  - Pessoal
  - Níveis Relacionados

Education & Guidance

EG 1

Offer development staff access to resources around the topics of secure programming and deployment

Activities

A. Conduct technical security awareness training

Either internally or externally sourced, conduct security training for technical staff that covers the basic tenets of application security. Generally, this can be accomplished via instructor-led training in 1-2 days or via computer-based training with modules taking about the same amount of time per developer.

Course content should cover both conceptual and technical information. Appropriate topics include high-level best practices surrounding input validation, output encoding, error handling, logging, authentication, authorization. Additional coverage of commonplace software vulnerabilities is also desirable such as a Top 10 list appropriate to the software being developed (web applications, embedded devices, client-server applications, back-end transaction systems, etc.). Wherever possible, use code samples and lab exercises in the specific programming language(s) that applies.

To rollout such training, it is recommended to mandate annual security training and then hold courses (either instructor-led or computer-based) as often as required based on development head-count.

B. Build and maintain technical guidelines

For development staff, assemble a list of approved documents, web pages, and technical notes that provide technology-specific security advice. These references can be assembled from many publicly available resources on the Internet. In cases where very specialized or proprietary technologies permeate the development environment, utilize senior, security-savvy staff to build security notes over time to create such a knowledge base in an ad hoc fashion.

Ensure management is aware of the resources and briefs oncoming staff about their expected usage. Try to keep the guidelines lightweight and up-to-date to avoid clutter and irrelevance. Once a comfort-level has been established, they can be used as a qualitative checklist to ensure that the guidelines have been read, understood, and followed in the development process.

Results

- Increased developer awareness on the most common problems at the code level
- Maintain software with rudimentary security best-practices in place
- Set baseline for security know-how among technical staff
- Enable qualitative security checks for baseline security knowledge

Success Metrics

- >50% development staff briefed on security issues within past 1 year
- >75% senior development/ architect staff briefed on security issues within past 1 year
- Launch technical guidance within 3 months of first training

Costs

- Training course buildout or license
- Ongoing maintenance of technical guidance

Personnel

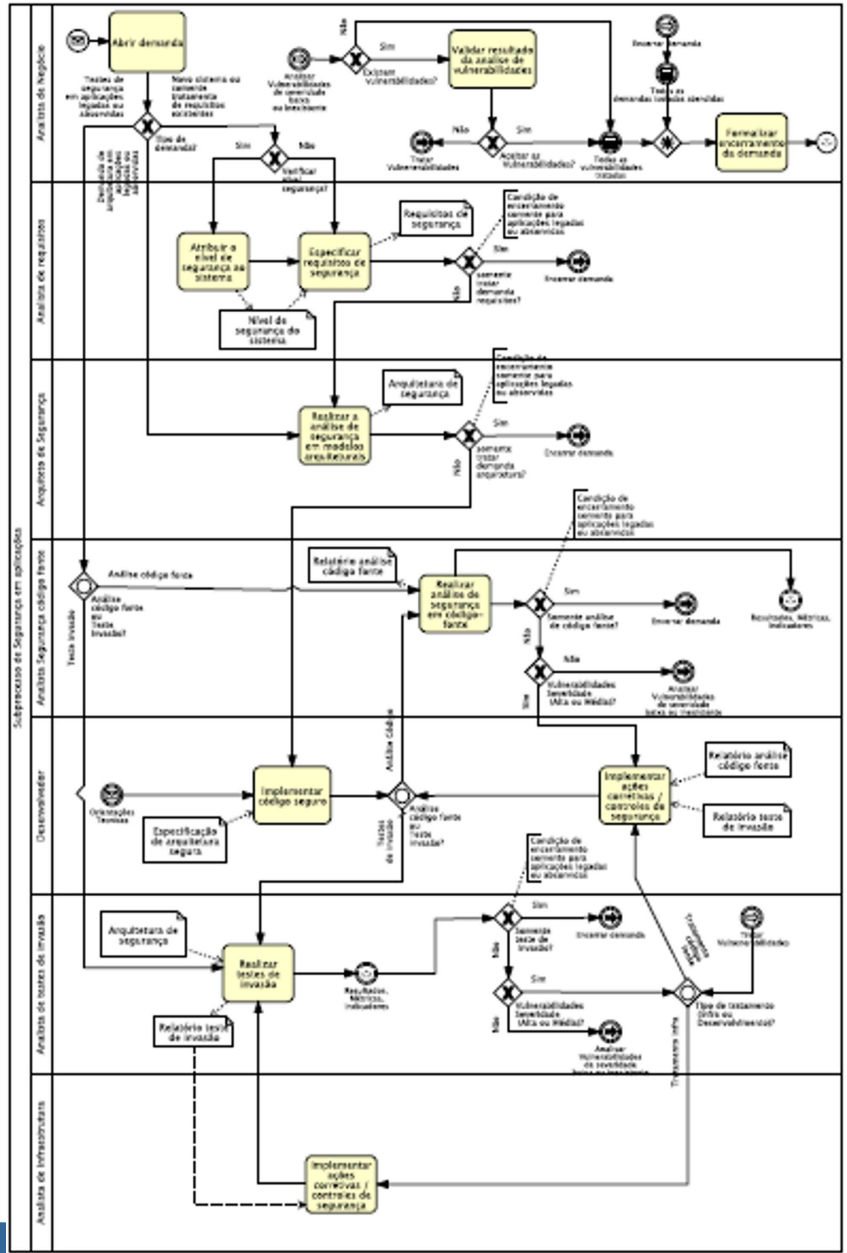
- Developers (1-2 days/yr)
- Architects (1-2 days/yr)

Related Levels

- Policy & Compliance - 2
- Security Requirements - 1
- Secure Architecture - 1

Desenho do Processo

- Visão geral do modelo



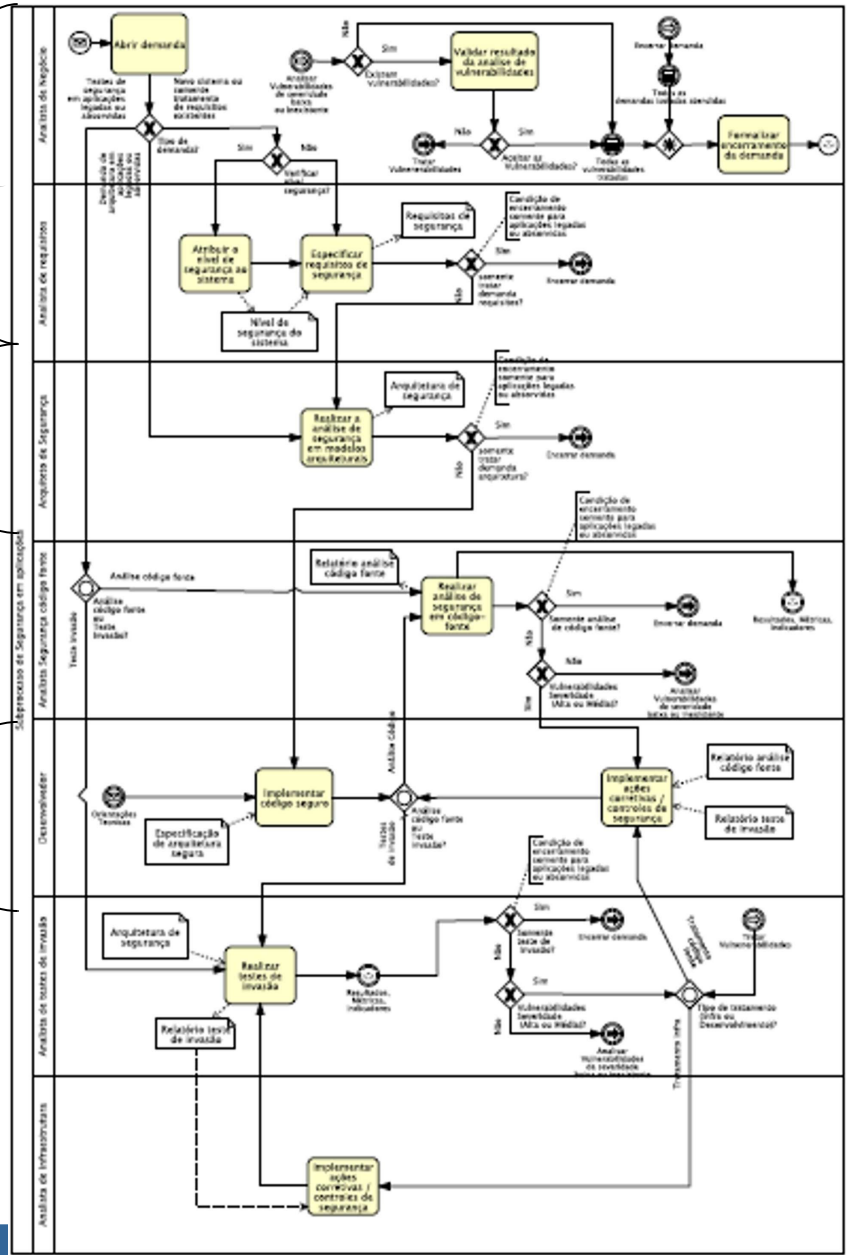
Desenho do Processo

- Visão geral do modelo

Engenharia de requisitos segura

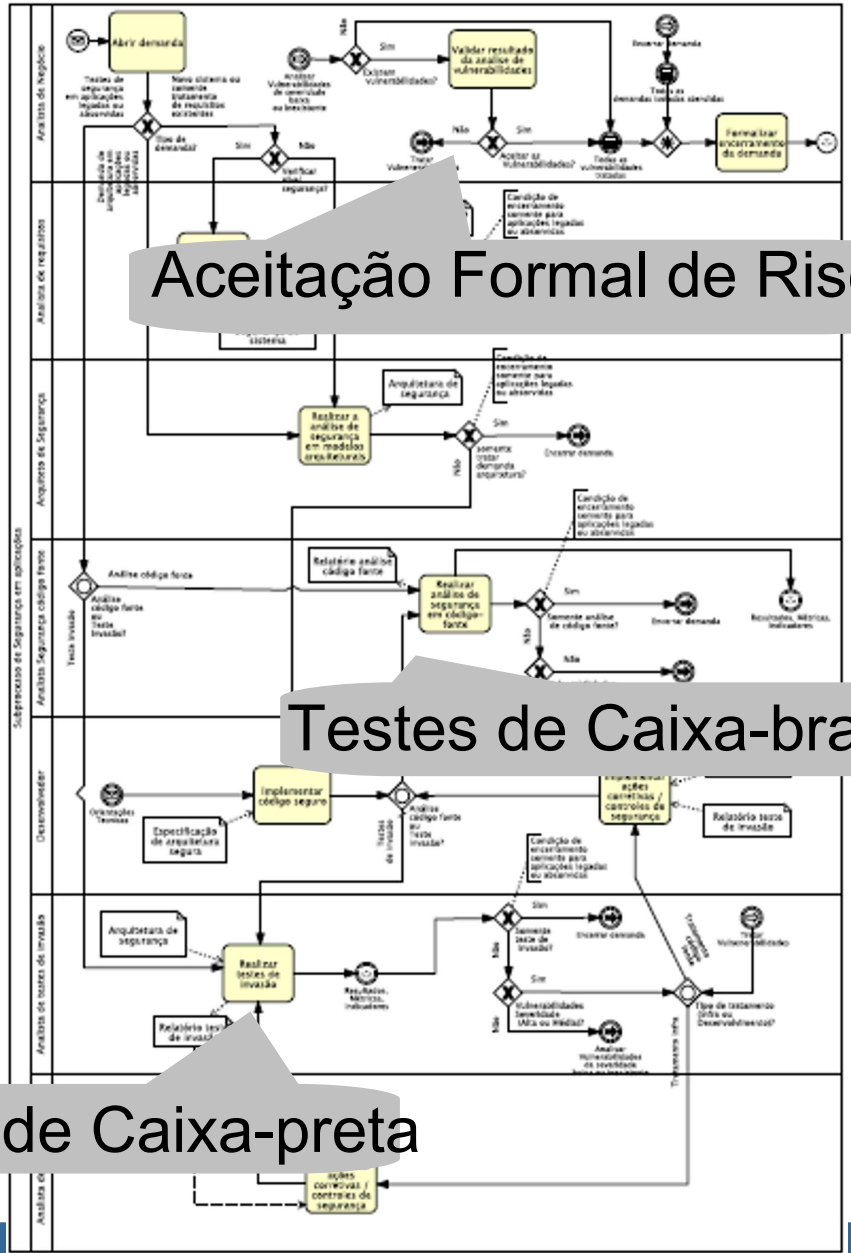
Segurança Arquitetural

Implementação segura



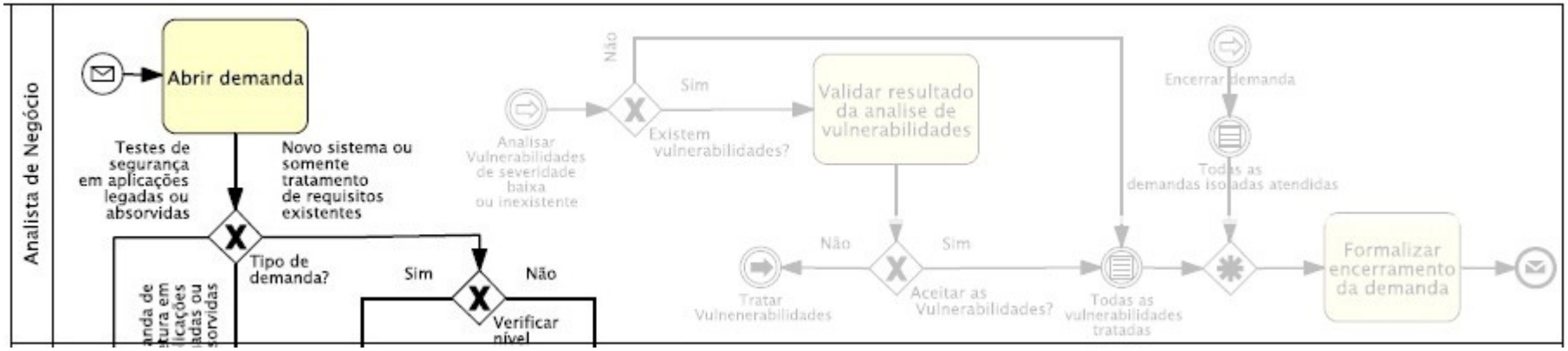
Desenho do Processo

- Visão geral do modelo



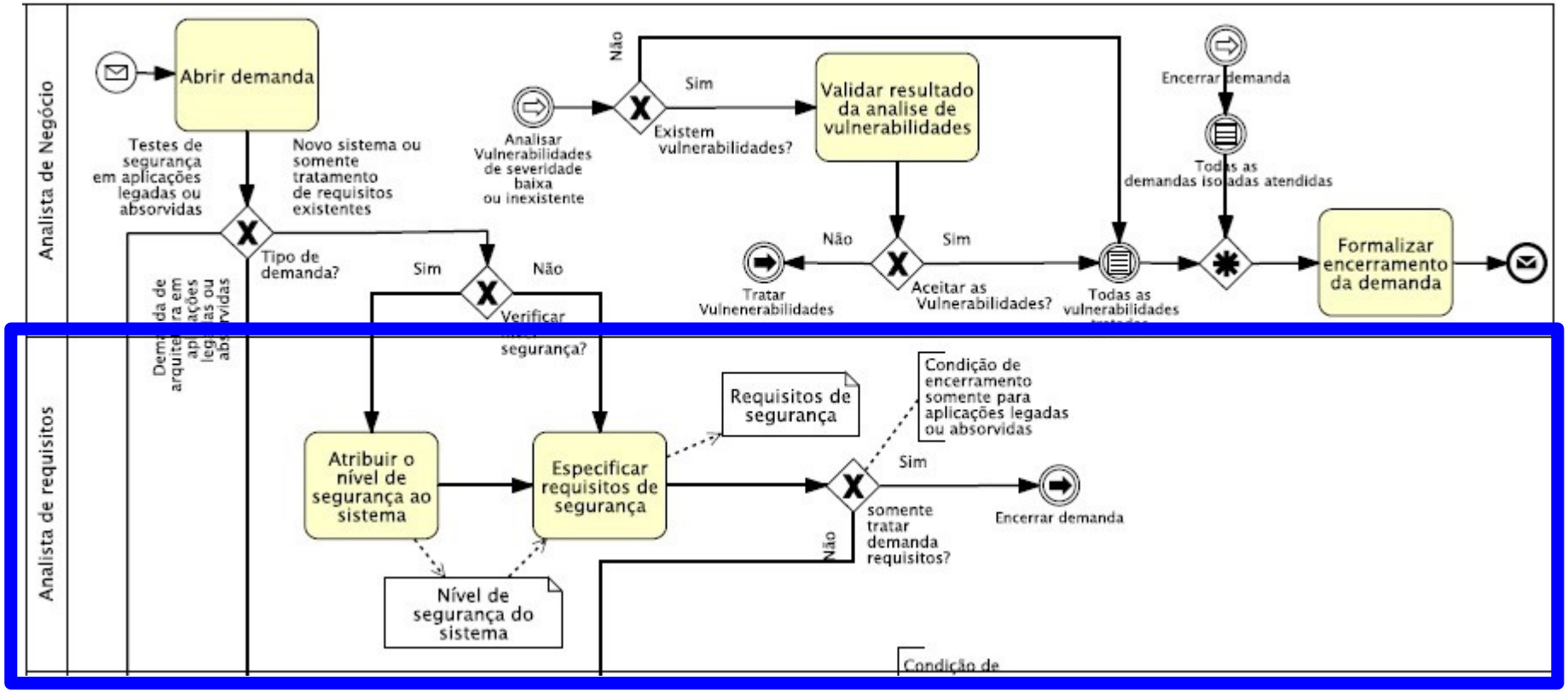
Desenho do Processo

- Engenharia de requisitos segura



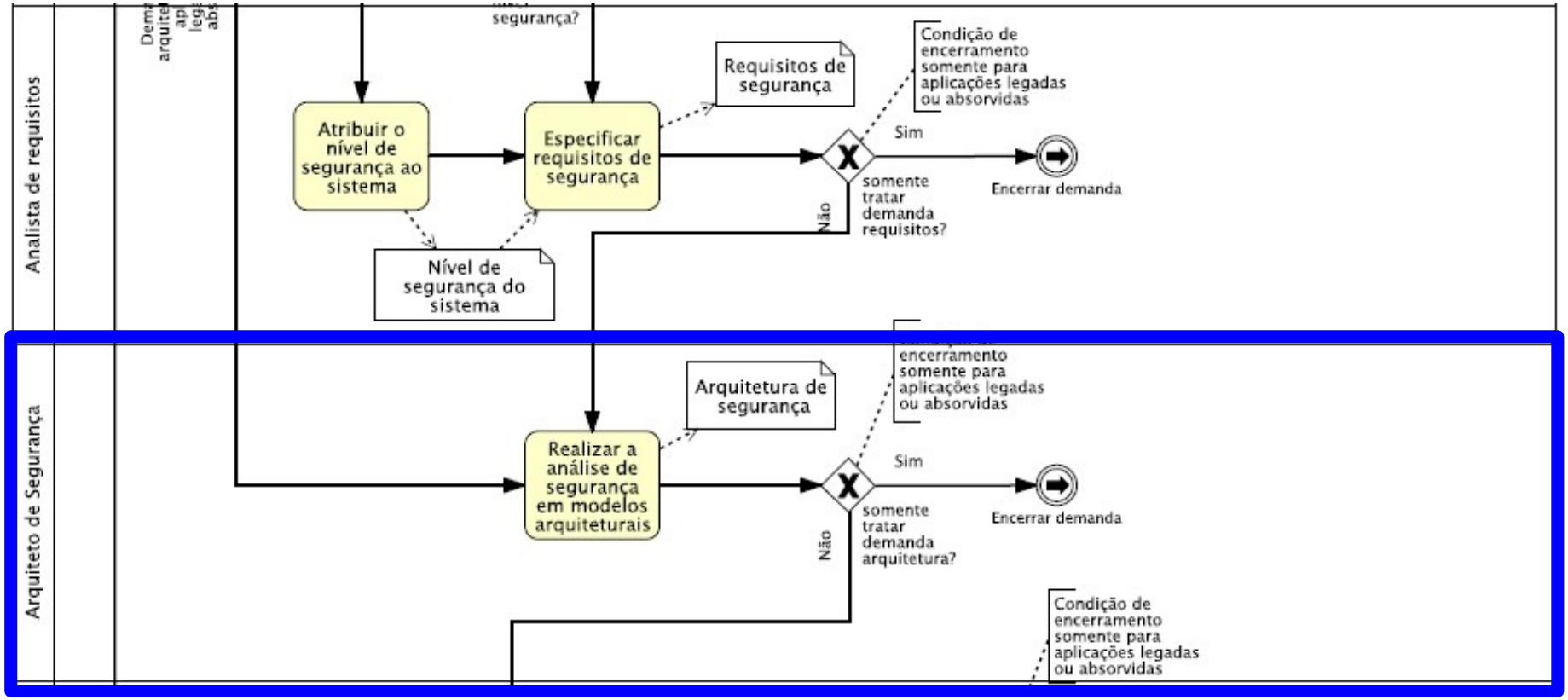
Desenho do Processo

- Engenharia de requisitos segura



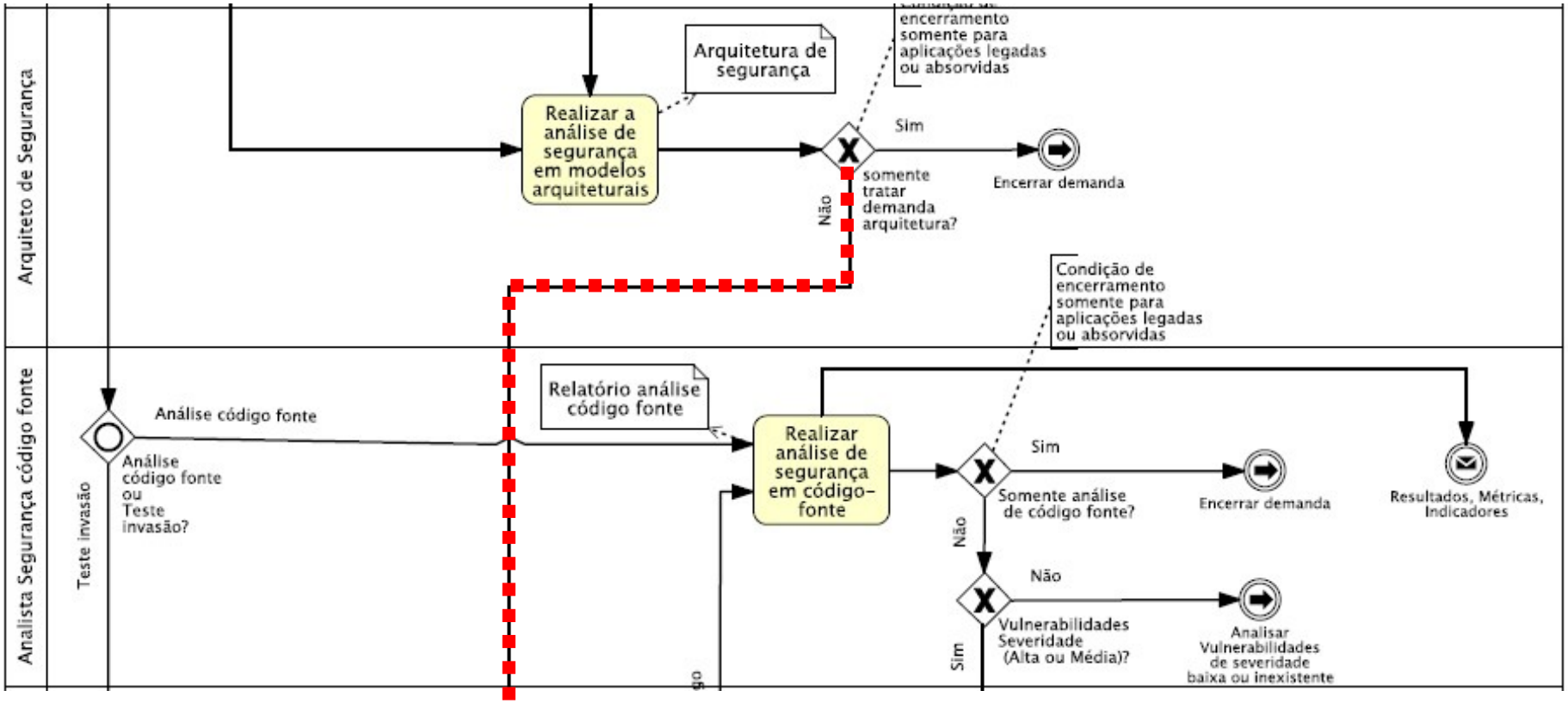
Desenho do Processo

- Segurança Arquitetural



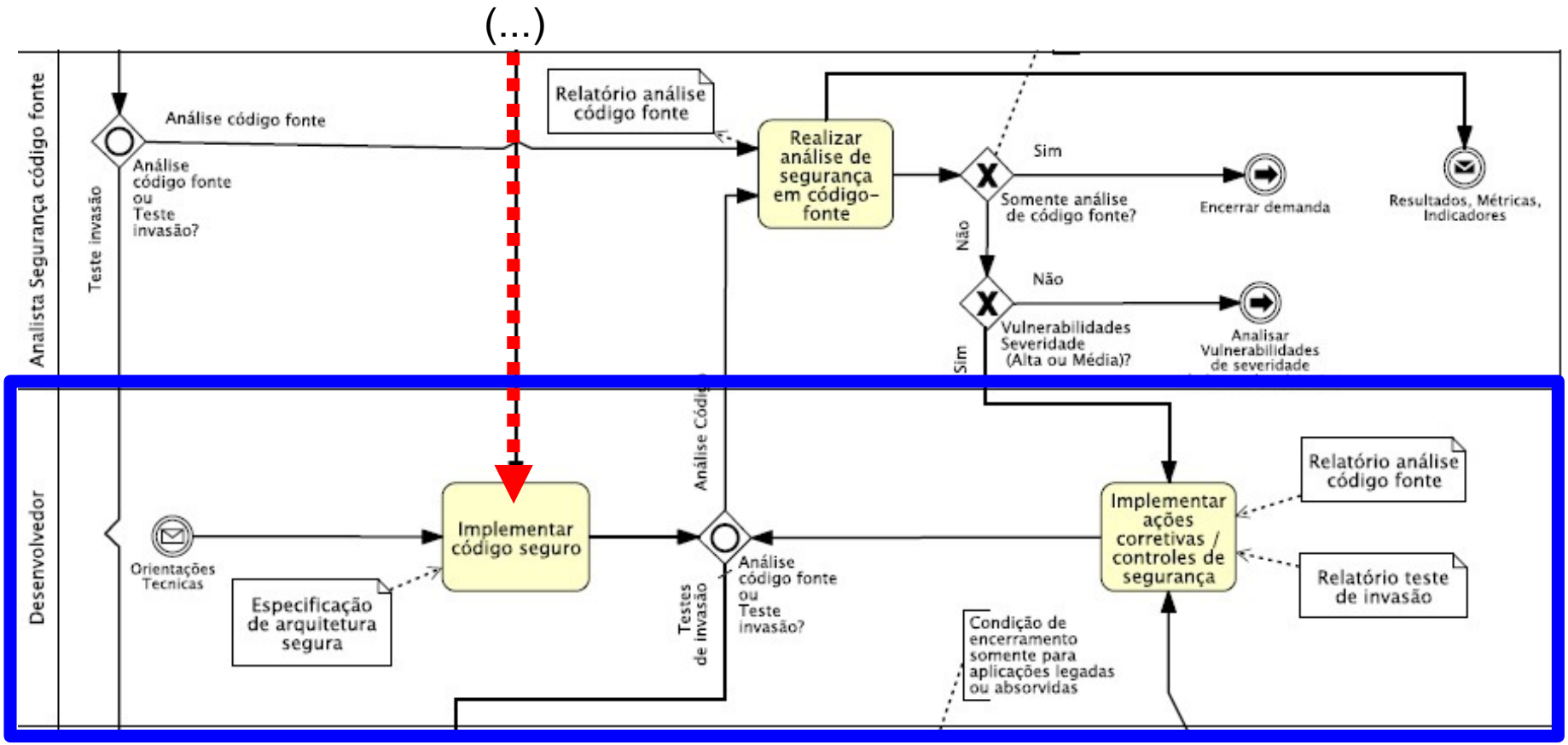
Desenho do Processo

- Implementação segura



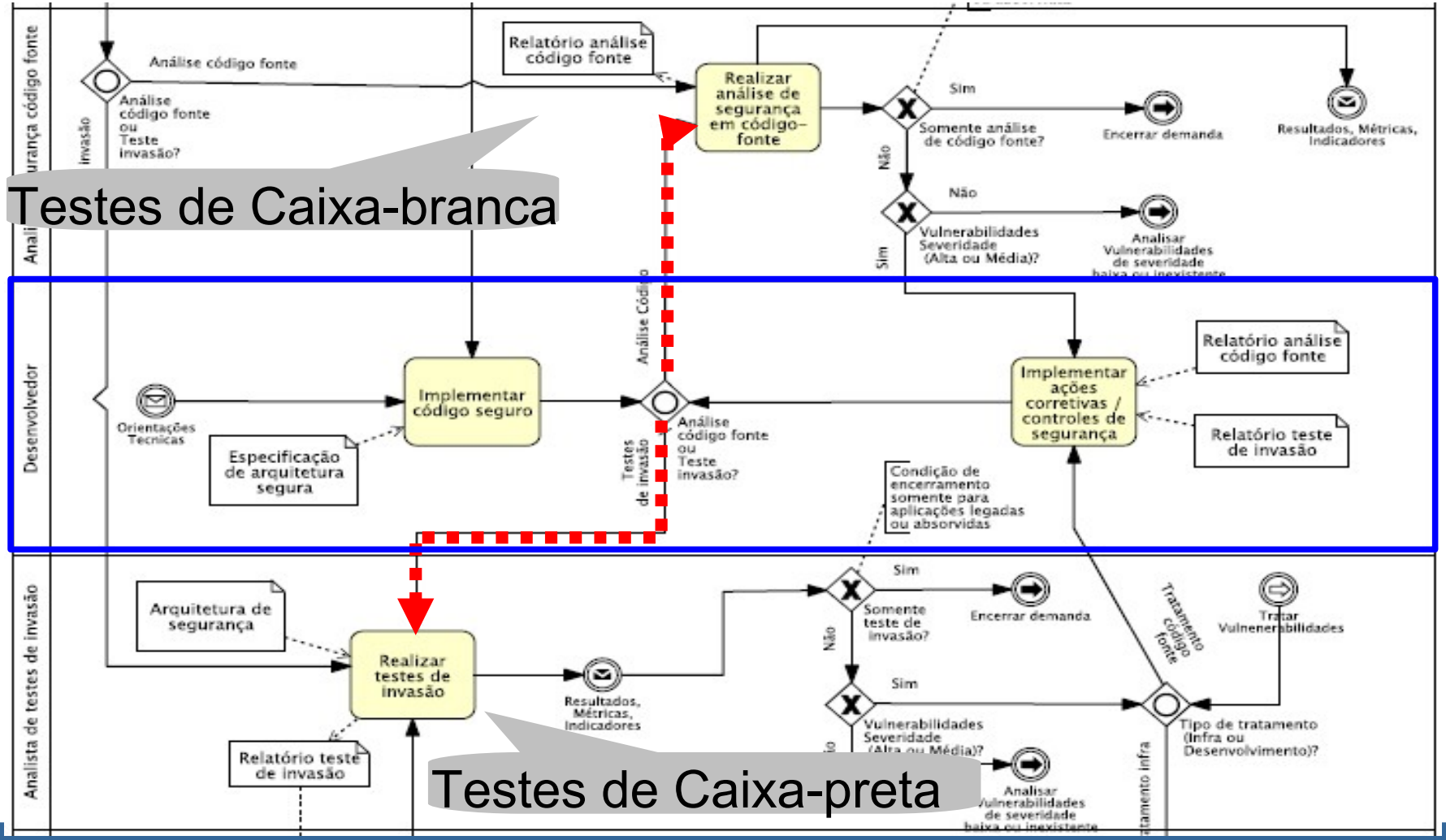
Desenho do Processo

- Implementação segura



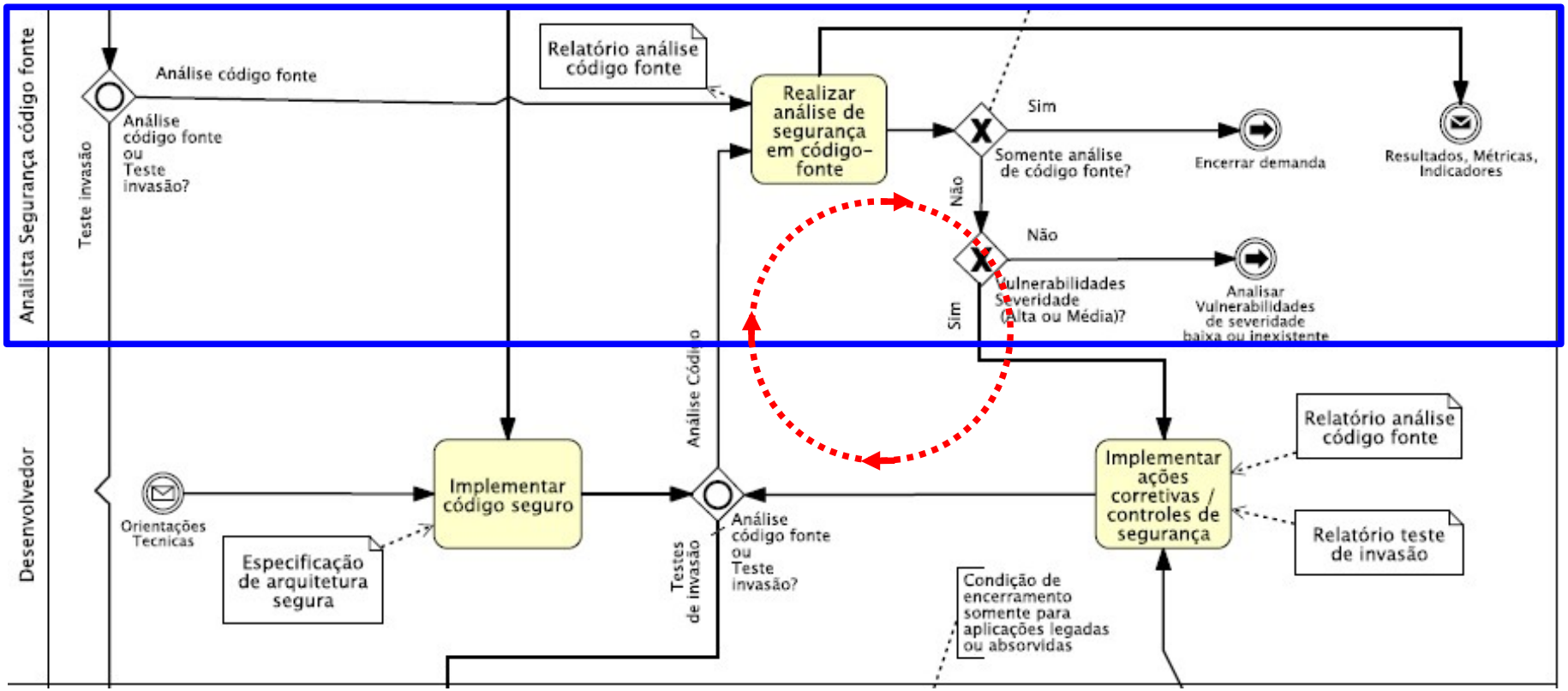
Desenho do Processo

- Implementar código seguro



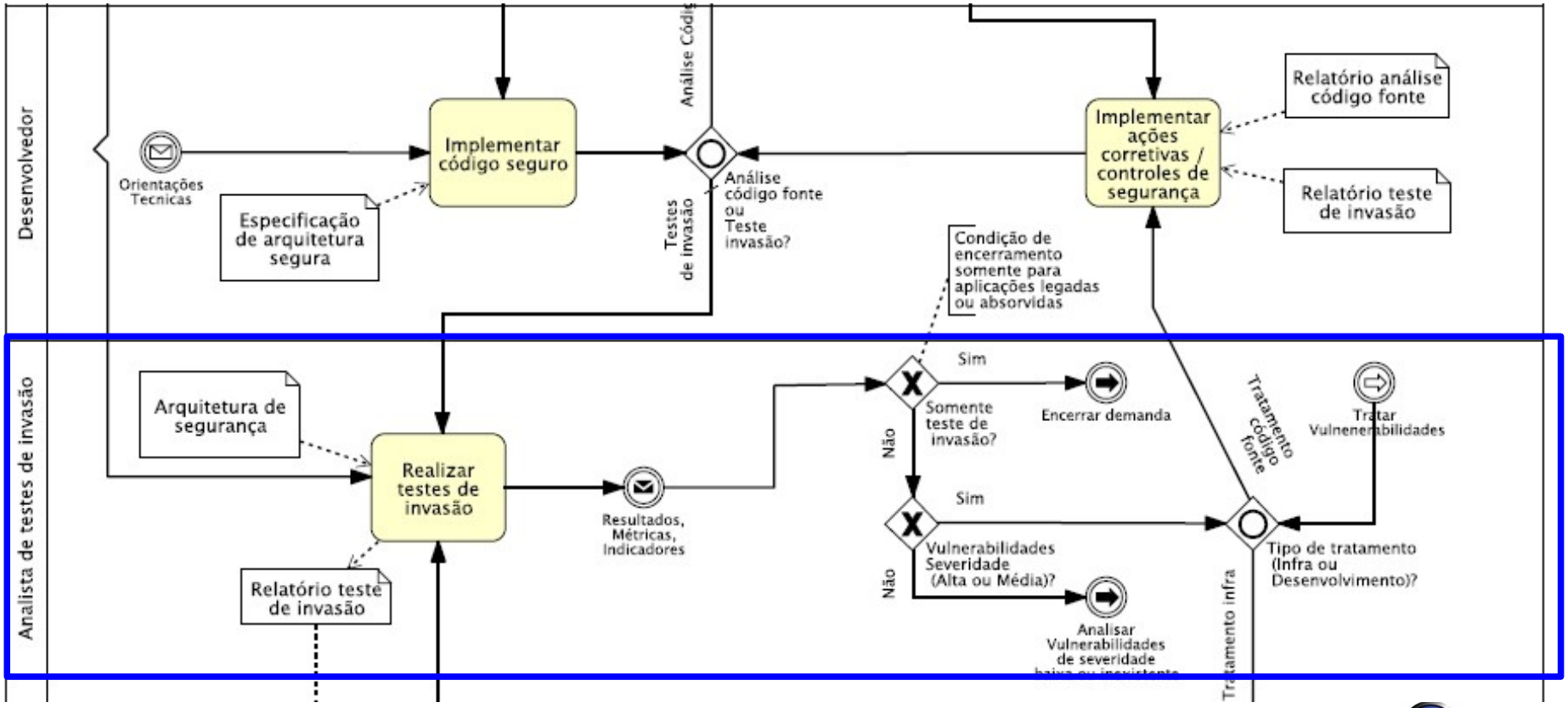
Desenho do Processo

- Testes de Caixa-branca
  - Implementar ações corretivas no código-fonte



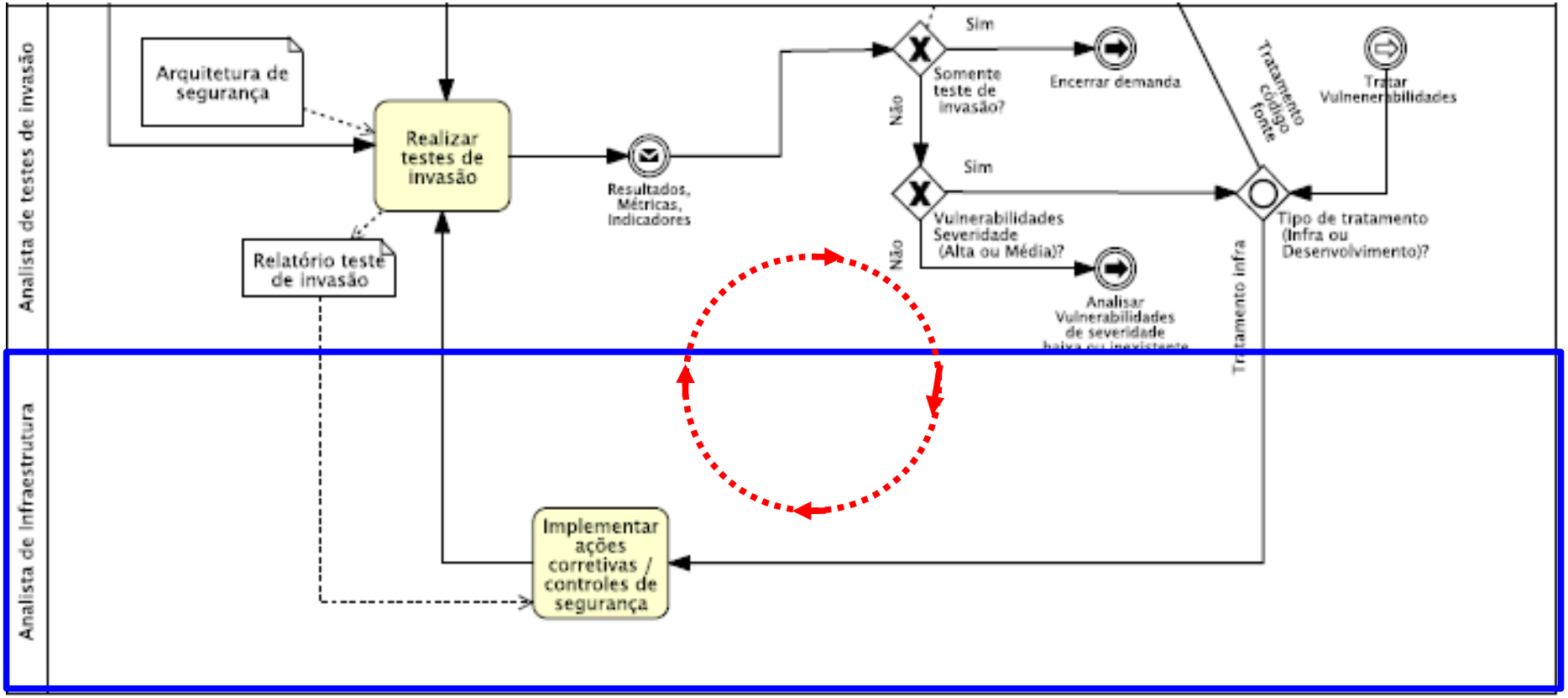
Desenho do Processo

- Testes de Caixa-preta



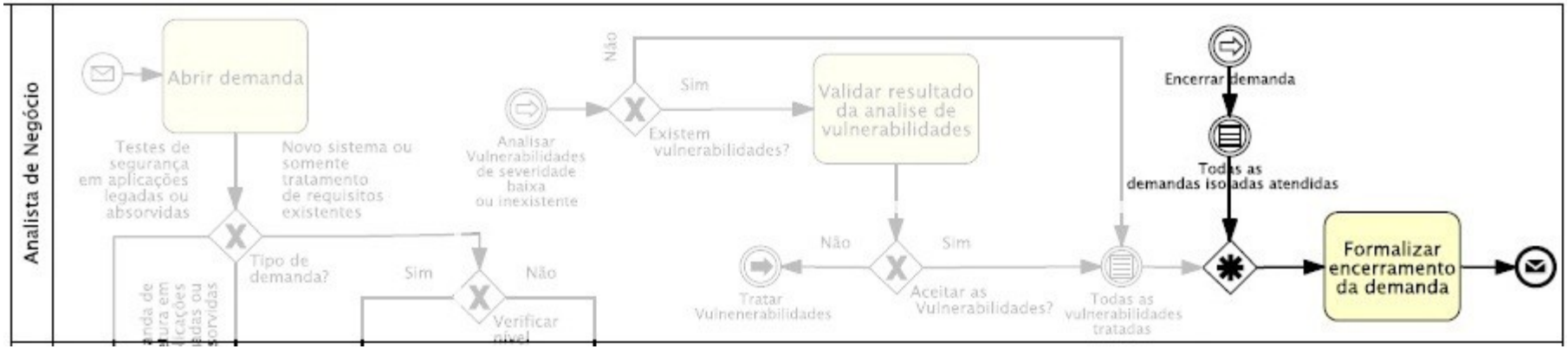
Desenho do Processo

- Testes de Caixa-preta
  - Implementar ações corretivas de infraestrutura



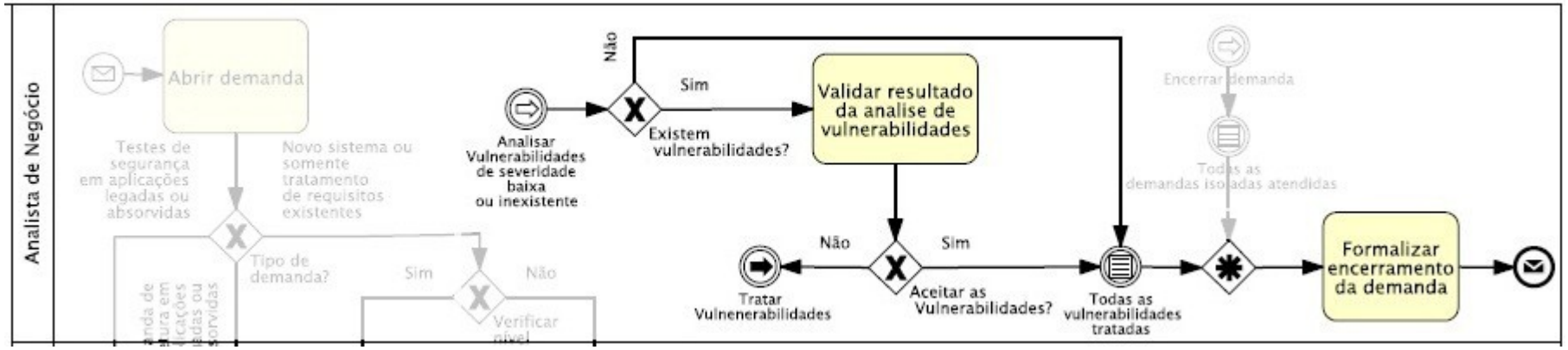
Desenho do Processo

- Encerrar demanda



Desenho do Processo

- Aceitação Formal de Riscos



Conclusões

- Os problemas de segurança em software, especialmente aplicações web, constituem grande fator de risco aos sistemas de informação das organizações.
- O nível de conhecimento de boa parte dos desenvolvedores que temos no mercado é insuficiente.

Conclusões

- As empresas que produzem software precisam melhorar os processos de desenvolvimento de software existentes, por meio de introdução de práticas, como as propostas pelo OWASP.
- Todos devem estar convencidos de que as preocupações que envolvem a segurança de aplicações **não devem estar restritas apenas** às equipes de desenvolvimento/testes, mas todos os envolvidos no ciclo de vida de desenvolvimento.

Duvidas?

Obrigado!

Referências

Maristela T. de Holanda; Jorge Henrique C. Fernandes, Jorge Henrique Cabral, SEGURANÇA NO DESENVOLVIMENTO DE APLICAÇÕES, Gestão da Segurança da Informação e Comunicações, UNB, 2011.

Pravir Chandra, OpenSAMM, Software Assurance Maturity Model, http://www.opensamm.org.

https://www.owasp.org/index.php/Category:OWASP_CLASP_Project