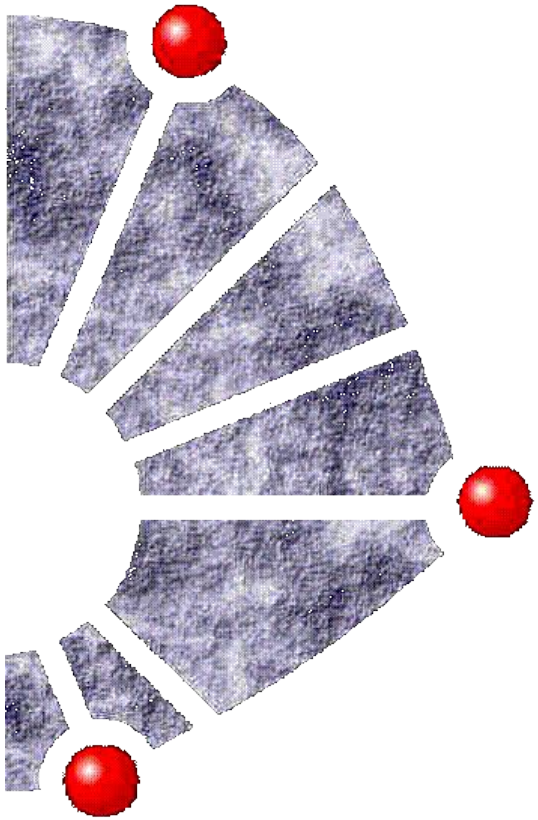
O Processo Unified e a Fase de Implementação e Testes



FEEC-UNICAMP

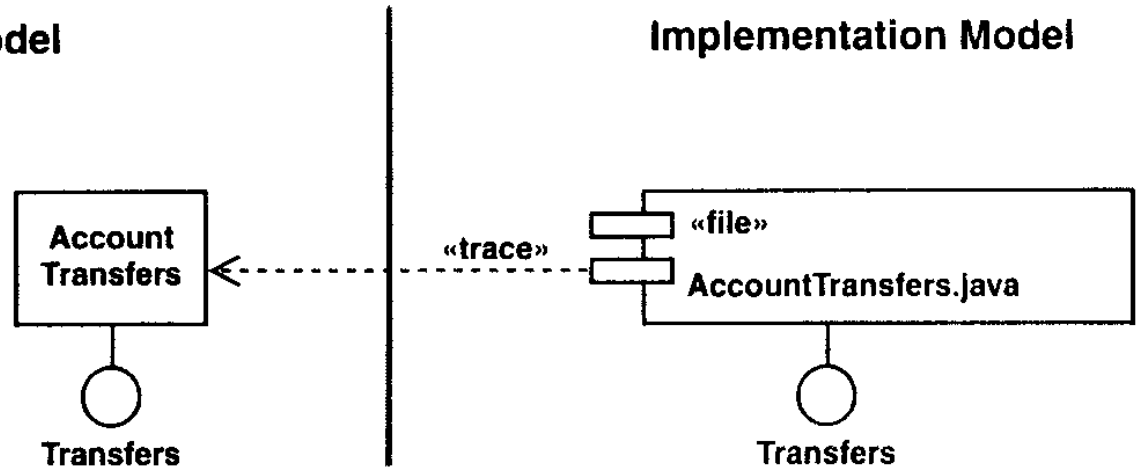
Ricardo Gudwin

Implementação

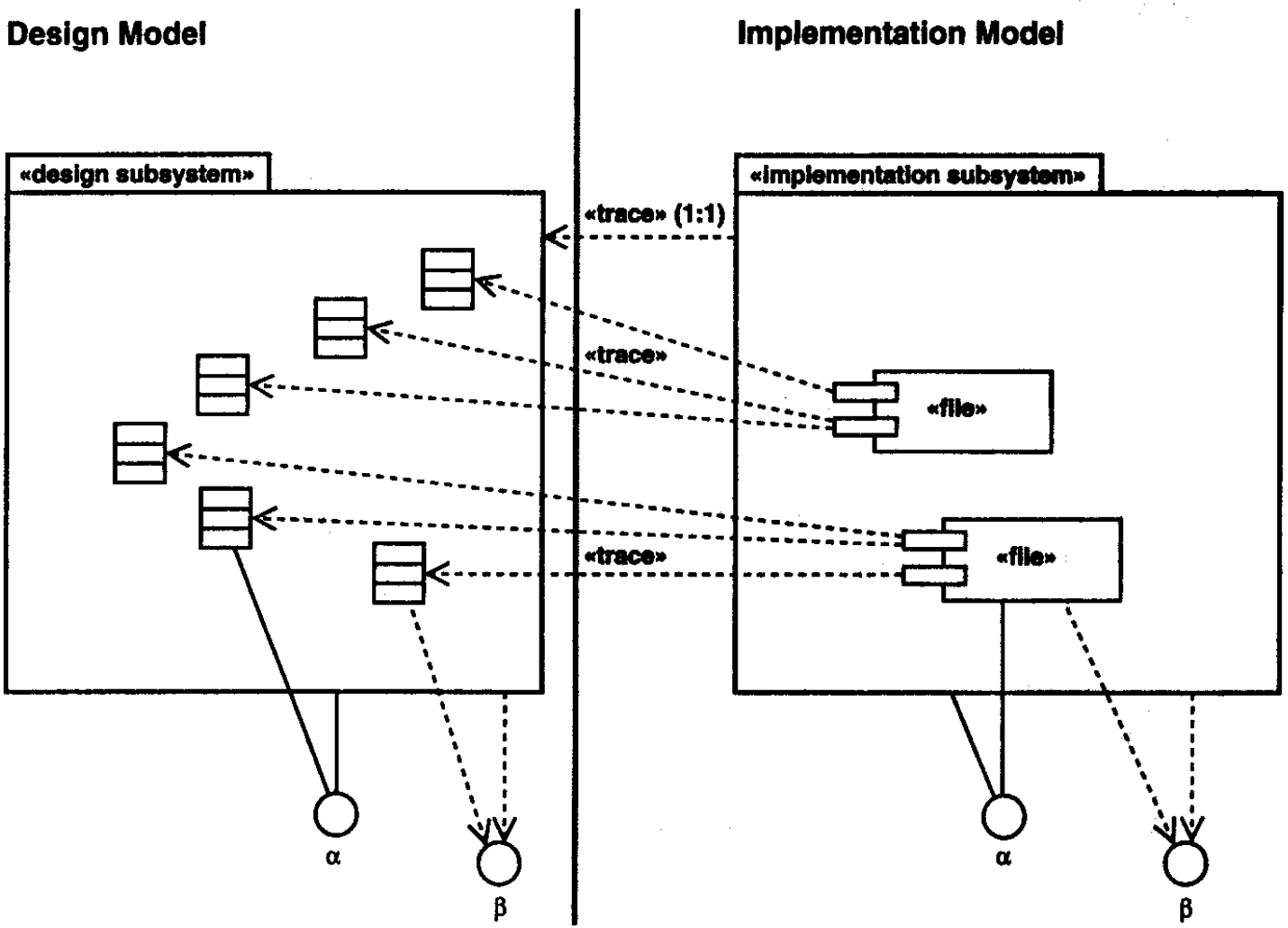
- Implementação
  - utiliza-se os artefatos desenvolvidos no design para implementar o sistema em termos de seus componentes, ou seja, código fonte, scripts, binários, executáveis, etc ...
- Objetivos
  - planejar a integração do sistema necessária na iteração corrente
  - distribuir o sistema entre componentes executáveis localizados nos nós do modelo de distribuição
  - implementação das classes de design e dos subsistemas especificados no design (geração de código)
  - teste individual dos componentes e posterior integração em módulos executáveis

Implementação

- Componentes
  - possuem diversos estereótipos
    - «executable», «file», «library», «table», «document»
  - possuem relacionamentos do tipo «trace» com os elementos do modelo que implementam
  - podem implementar individualmente diversos modelos diferentes

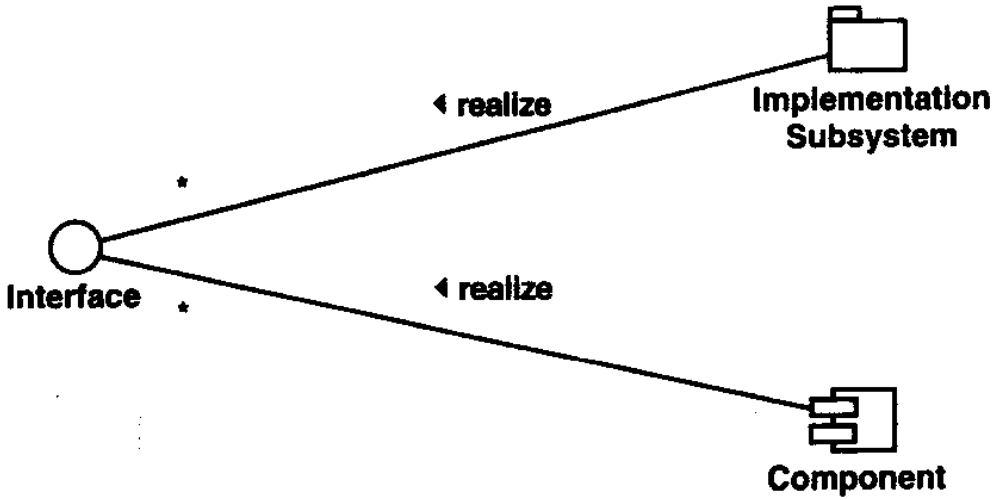


Implementação

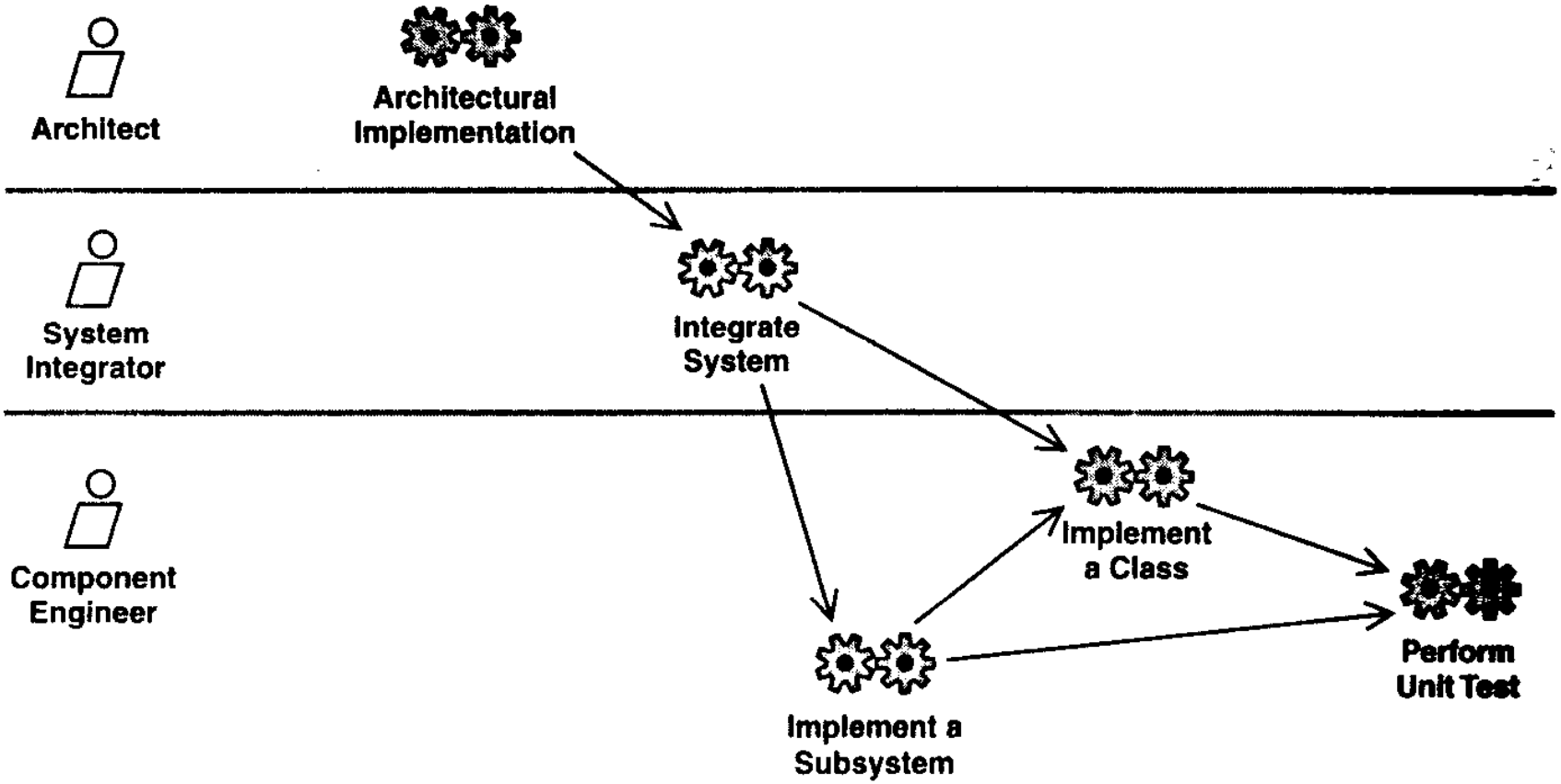


Implementação

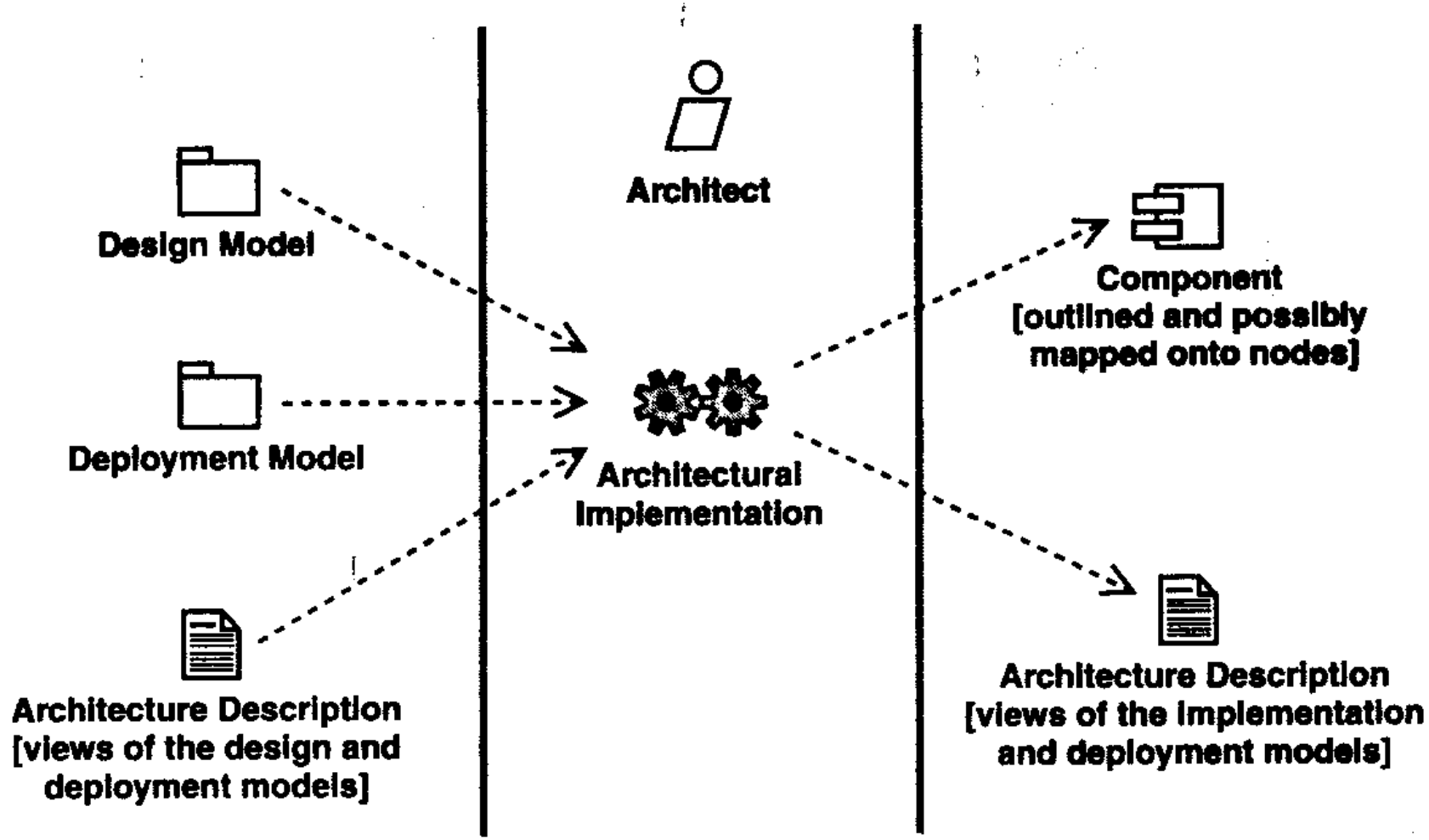
- Subsistemas de Implementação
  - package em Java
  - project em VisualBasic
  - diretório de arquivos em um projeto C++
  - subsistema em IDEs tais como o Rational Apex
  - component view package, em ferramentas de modelagem visual tais como o Rational Rose



Implementação



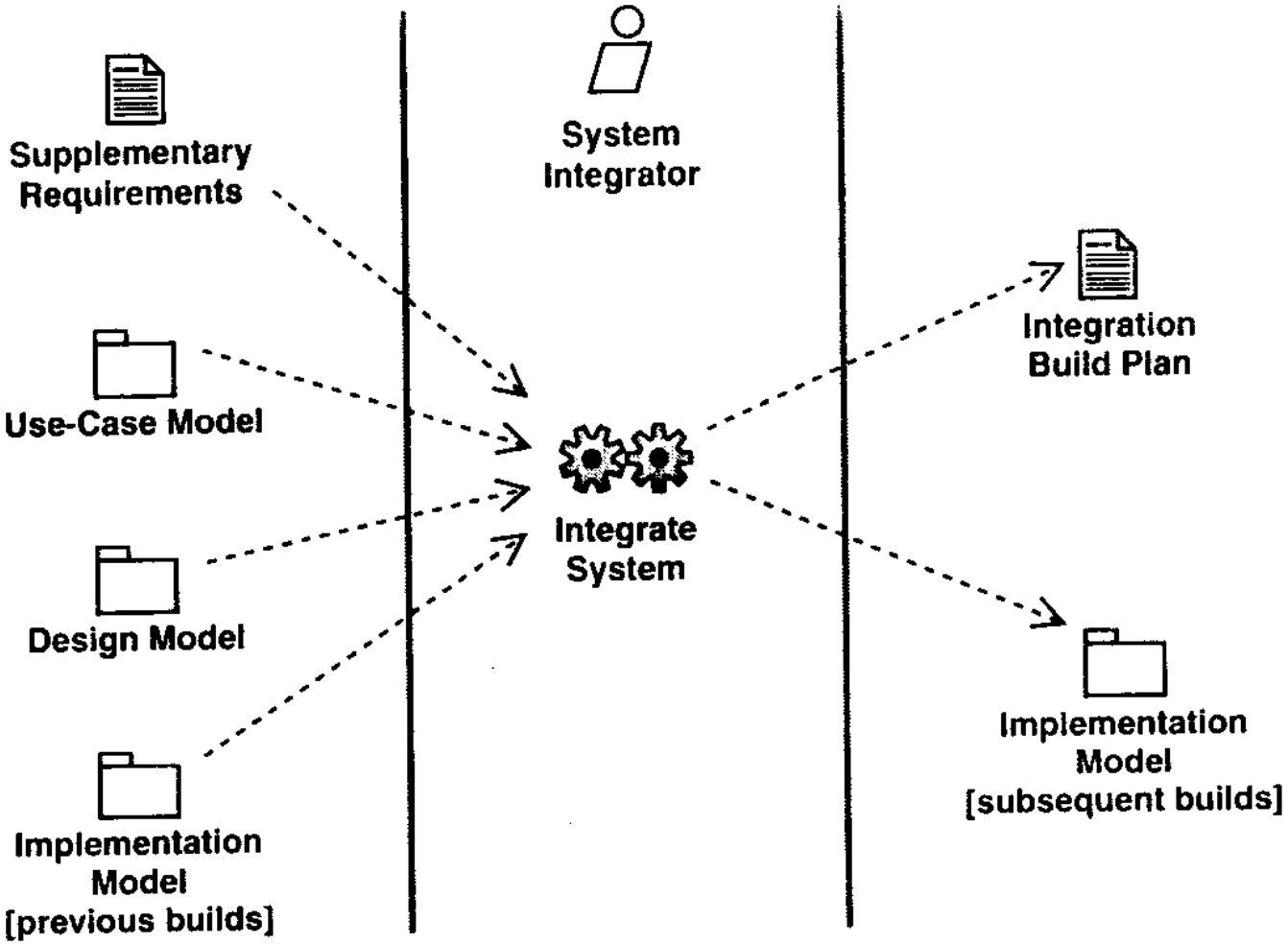
Implementação Arquitetural



Implementação Arquitetural

- Objetivo
  - criar um outline do modelo de implementação
    - identificação de componentes arquiteturalmente significativos, tais como componentes executáveis
    - mapeamento desses componentes em nós da configuração de rede
- Identificação de Componentes arquiteturalmente significativos
  - diagrama de componentes
- Identificação dos Componentes Executáveis e seu Mapeamento nos Nós
  - diagrama de distribuição, mostrando onde se localizam
    - componentes
    - objetos ativos (aqueles com Thread de controle individual)

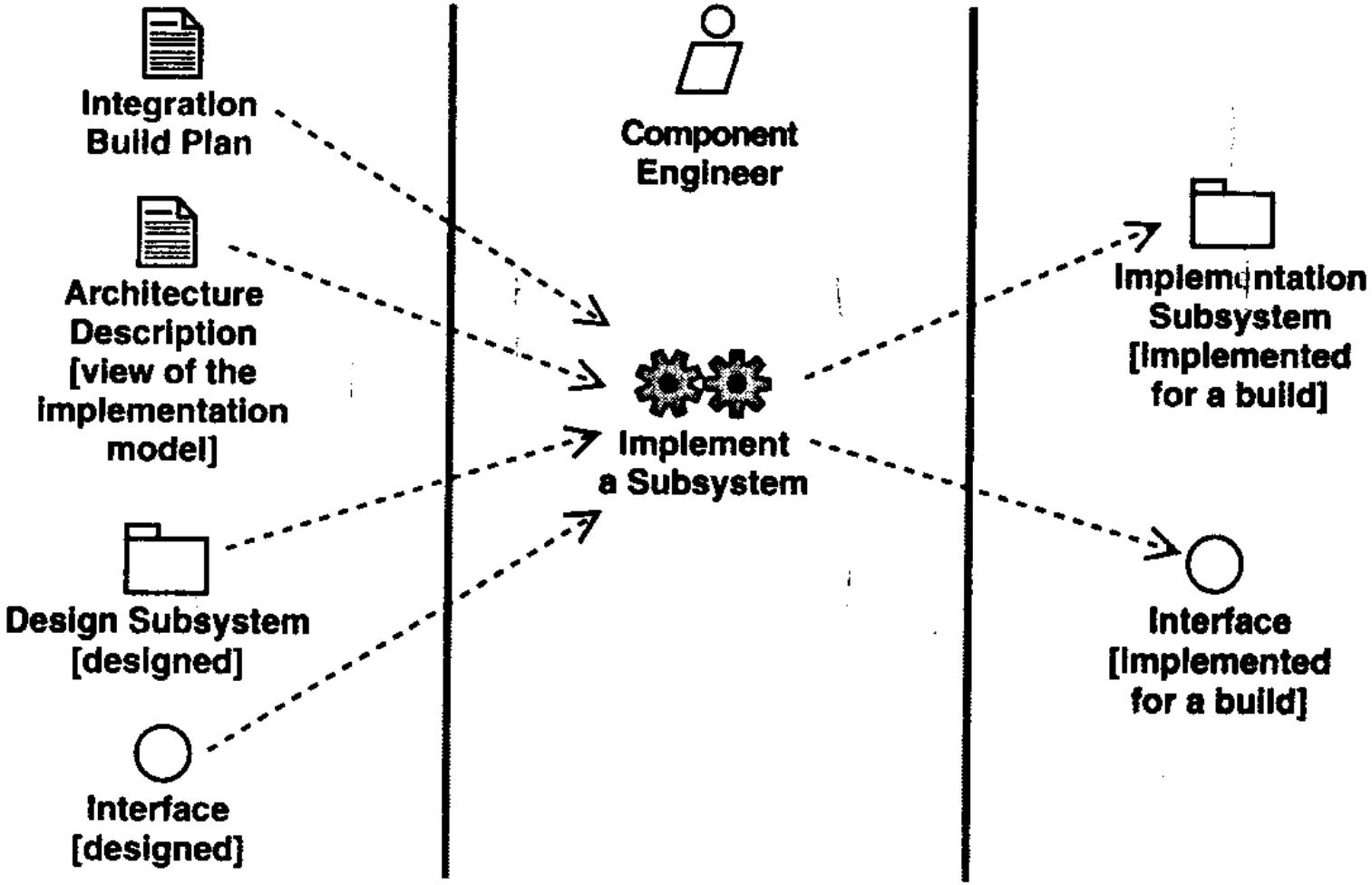
Integração do Sistema



Integração do Sistema

- Objetivos
  - criação da um plano de construção integrado, descrevendo as construções necessárias na iteração corrente e quais os requisitos que essa construção implementa
  - definição efetiva de quais componentes serão integrados e criação de scripts de compilação e linkagem
- Planejamento da Construção
  - verificação de implementações de iterações anteriores e eventual reuso de partes já consolidadas
  - adição de funcionalidades para a construção corrente, por meio da adição de novos casos de uso
- Integrando a Construção
  - criação de scripts de compilação e linkagem das versões corretas dos diferentes componentes sendo integrados

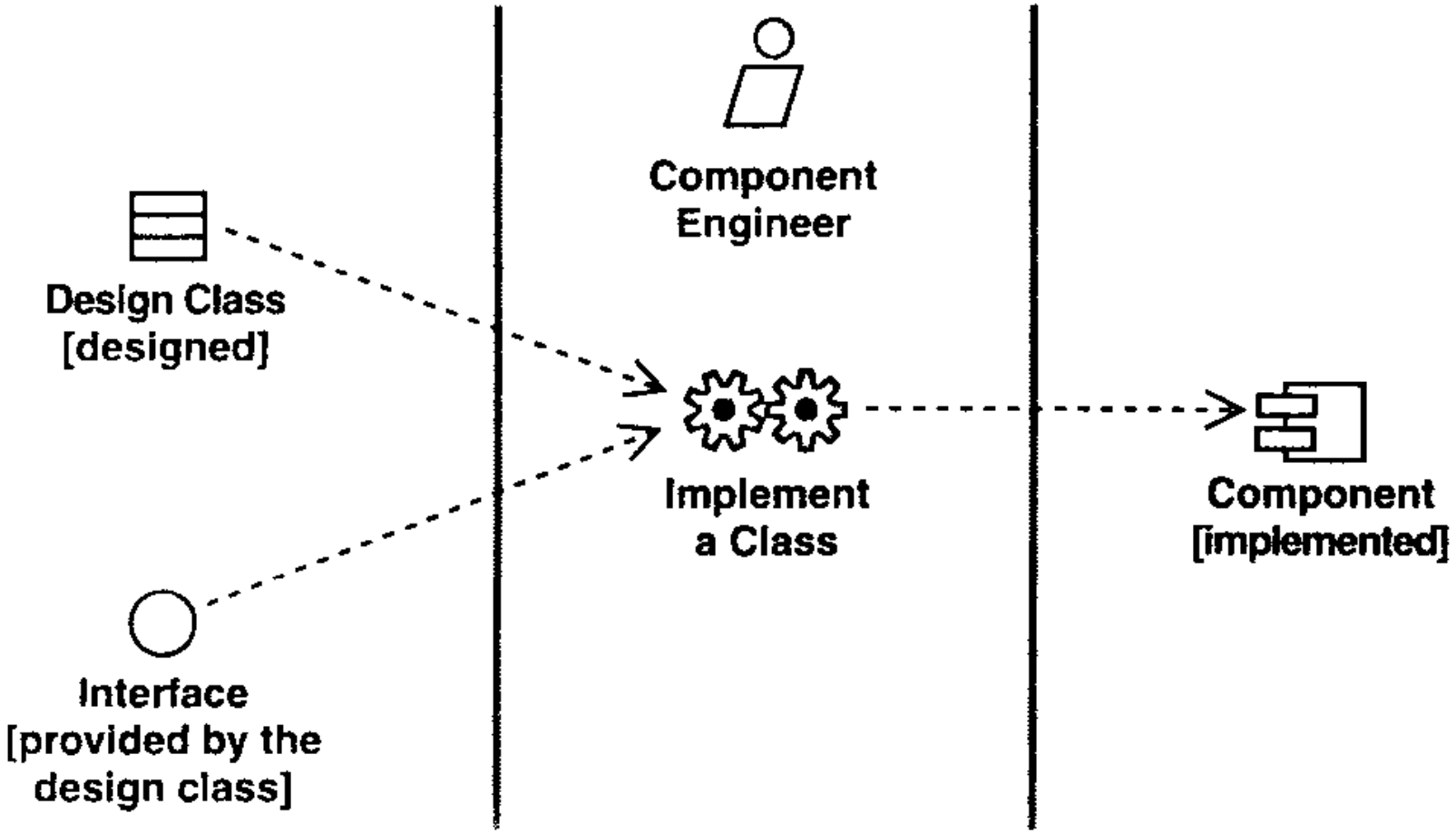
Implementação de Subsistemas



Implementação de Subsistemas

- Objetivos
  - garantir que a implementação de um subsistema cumpra seu papel, a cada construção, como estipulado pelo plano de construção integrada
    - garantir que os casos de uso e cenários estejam corretos e que as interfaces estejam corretas
- Gerenciamento do Conteúdo de Subsistemas
  - mesmo que o conteúdo de um subsistema já esteja determinado pelo arquiteto, ele pode requerer pequenos refinamentos implementados pelo engenheiro de componentes, a medida que a implementação se desenvolve
  - à medida que subsistemas contenham outros subsistemas recursivamente, a metodologia de mapeamento entre componentes e classes de design é análoga em cada nível

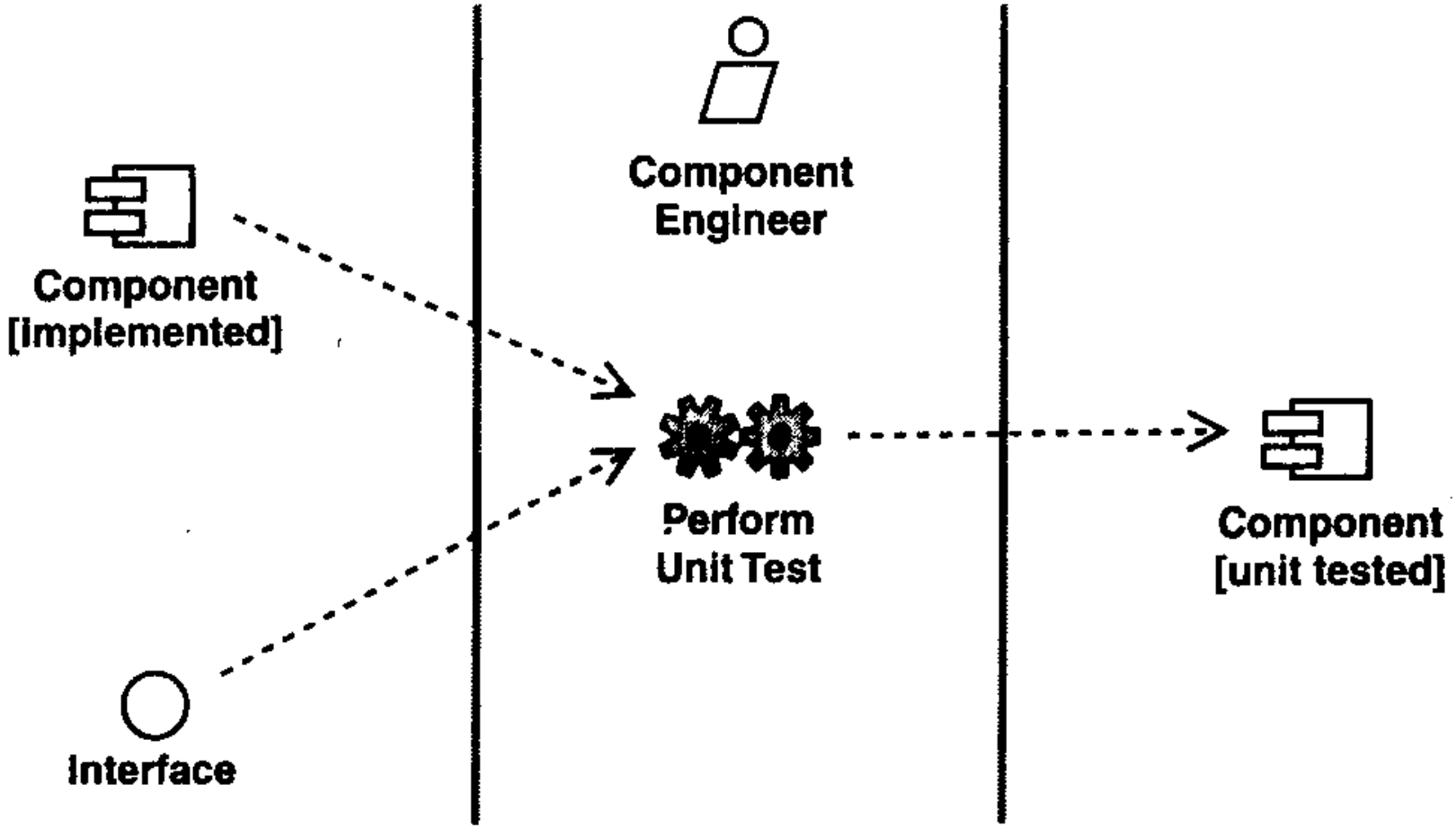
Implementando Classes



Implementando Classes

- Objetivos
  - implementação de classes de design na forma de componentes
    - distribuição do código fonte em um ou diversos arquivos
      - componentes do tipo «file»
    - geração do código fonte a partir das classes de design e dos relacionamentos em que participa
      - esqueletos de código podem ser gerados automaticamente
    - implementação das operações das classes de design em termos de métodos
      - alguns tipos de operações podem também ser geradas automaticamente, sendo complementadas com edição posterior
    - verificação de que o componente provê a mesma interface que a classe de design que implementa
    - eventualmente, conserto de defeitos, quando a classe já havia sido implementada em outras iterações

Teste de Unidade



Teste de Unidade

- Objetivos
  - realização de um teste individual do componente implementado, sem considerar sua posterior integração
- Tipos de Testes
  - Teste de Especificação
    - verifica o comportamento observável externo da unidade, sem entrar no mérito de como esse comportamento é implementado
    - focaliza nas entradas e saídas da unidade
  - Teste Estrutural
    - verifica a implementação interna da unidade, fazendo com que cada trecho do código seja executado
  - Outros Testes
    - testes de desempenho, uso de memória, escalabilidade e capacidade

Testes

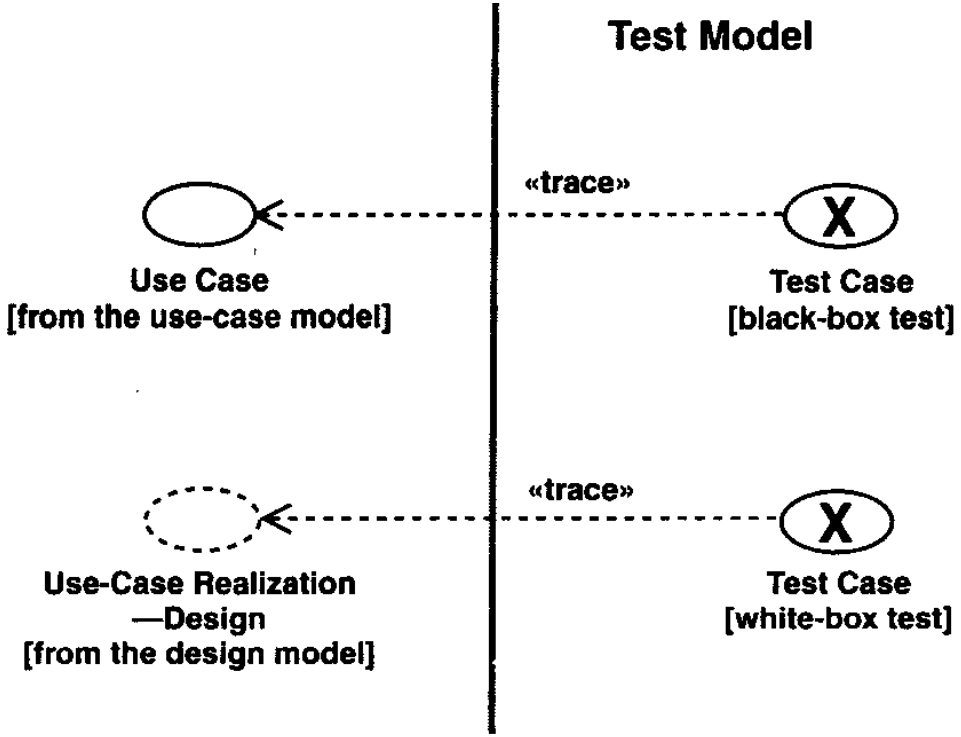
- Fase de Testes
  - verificação dos resultados da implementação por meio do teste de cada módulo do sistema e sua integração
- Objetivos
  - planejamento dos testes a serem executados em cada iteração
  - design e implementação dos testes por meio de casos de teste que especificam o que testar e quais os procedimentos a serem utilizados para os testes
  - criação de componentes de teste executáveis para a automação de testes, quando possível
  - avaliar sistematicamente os resultados de cada teste e encaminhar os módulos defectivos para que estes possam ser retrabalhados

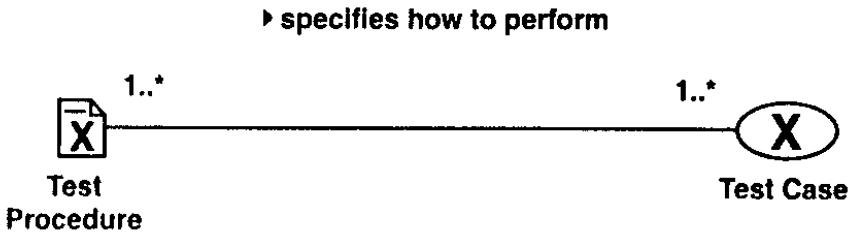
Testes

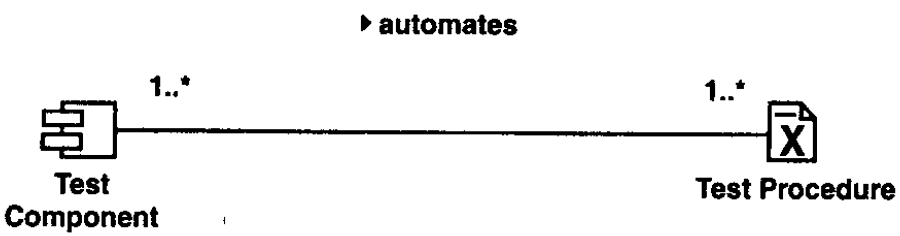
- Modelos de Teste
  - descreve como componentes executáveis do modelo de implementação são testados por meio de testes de integração e testes de sistema
  - descreve os aspectos específicos do sistema que serão testados
  - coleção de casos de teste, procedimentos de teste e componentes de teste
- Caso de Teste
  - corresponde a um caso de uso, só que com finalidades de teste
  - especifica um cenário de teste, incluindo o que testar, com que entradas e com quais resultados esperados
  - caso de teste pode estar associado a um caso de uso ou à realização do caso de uso no design

Testes

- Procedimentos de Teste
  - especifica como realizar um ou mais casos de teste
- Componente de Teste
  - automatizam um ou mais procedimentos de teste
  - linguagens de script ou linguagens de programação



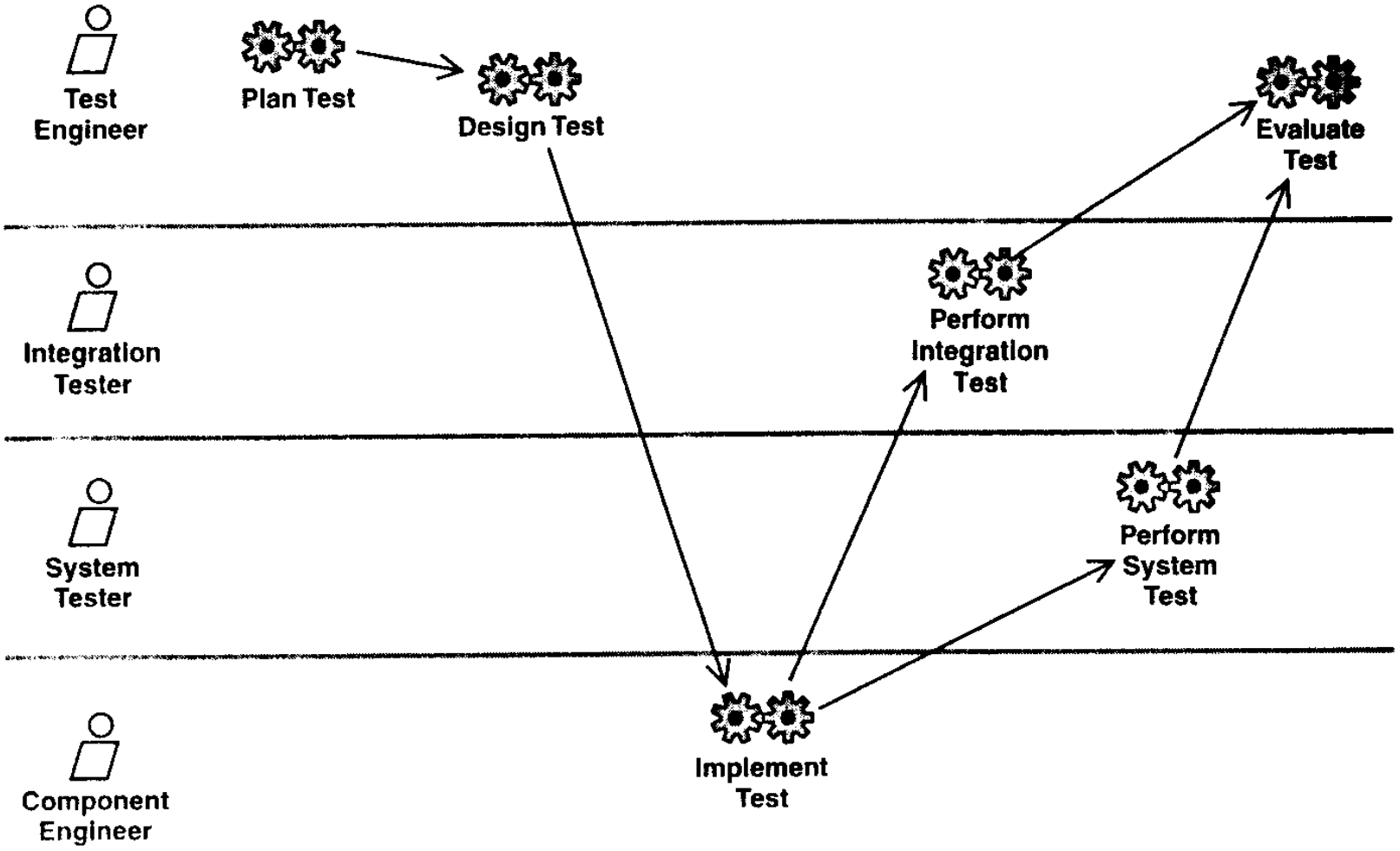




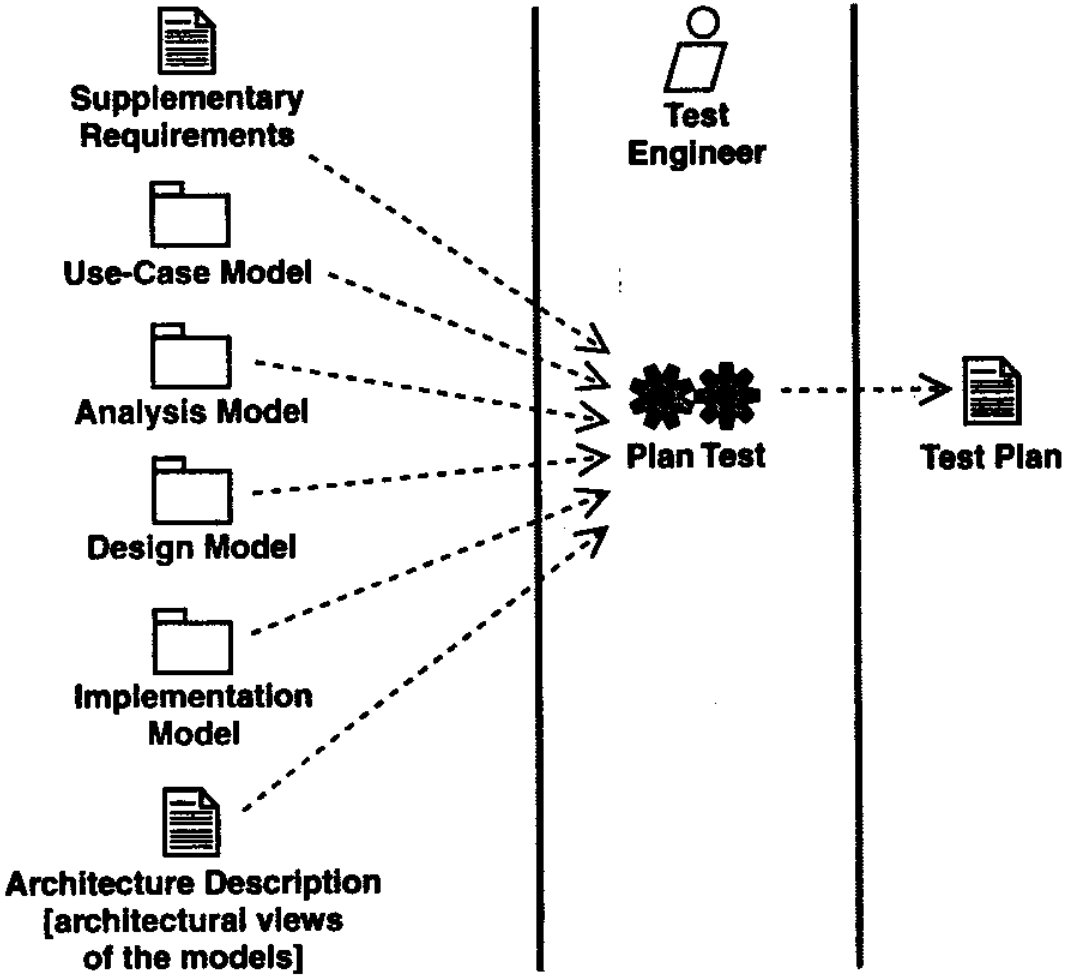
Testes

- Tipos de Teste
  - Testes de Instalação
    - verificam que o sistema pode ser instalado na plataforma do cliente e que o sistema opera corretamente após instalado
  - Testes de Configuração
    - verificam que o sistema opera corretamente sob diferentes configurações
  - Testes Negativos
    - tentam ocasionar falhas no sistema para revelar suas fraquezas
    - tenta-se utilizar o sistema de maneiras para as quais ele não foi projetado, e.g. testando-se configurações de rede defeituosas, hardware insuficiente ou demanda de uso (carga) intensiva
  - Testes de Stress
    - tentam verificar como o sistema se comporta quando os recursos disponíveis são insuficientes ou estão indisponíveis

Testes


Test Engineer
Plan Test
Design Test
Evaluate Test
Integration Tester
Perform Integration Test
System Tester
Perform System Test
Implement Test
Component Engineer

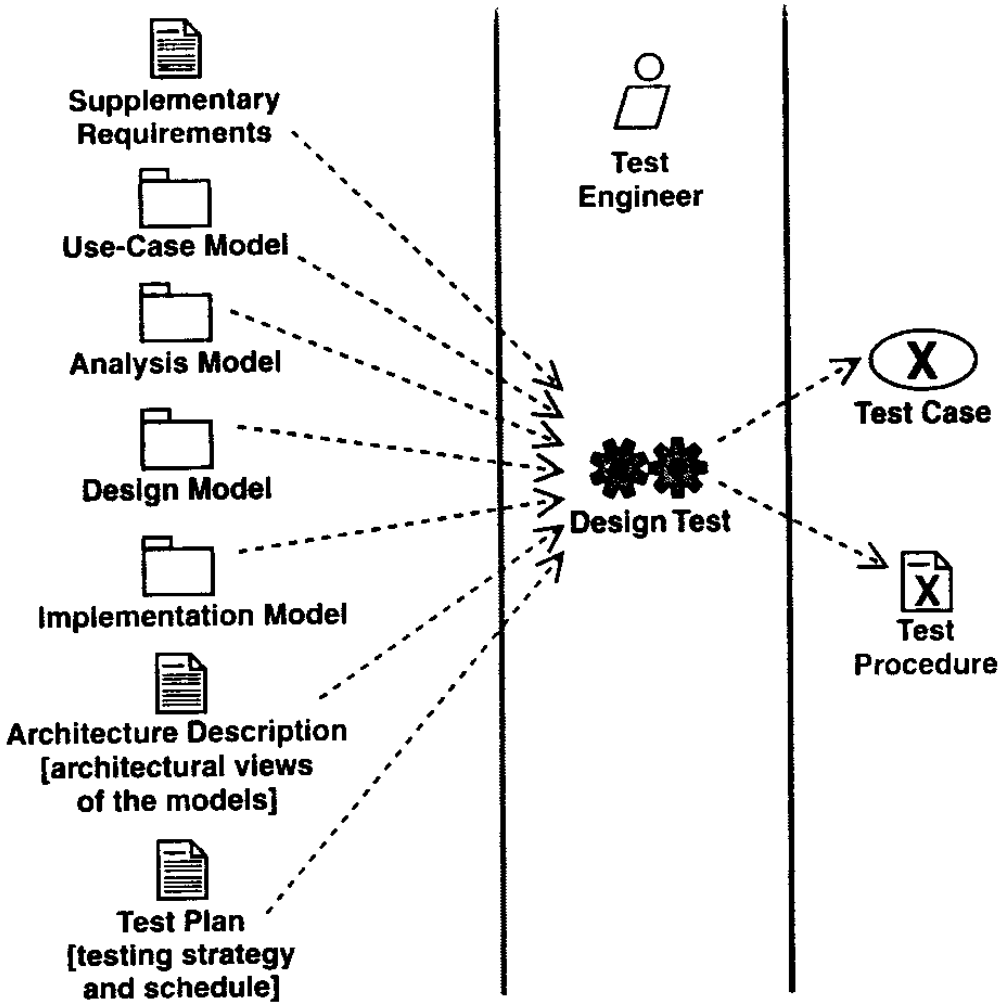
Planejamento de Testes



Planejamento de Testes

- Objetivos
  - planejar os esforços de teste dentro de uma iteração
    - descrevendo uma estratégia de testes
    - estimando os requisitos para o esforço de teste, tais como recursos humanos e do sistema
    - criando uma escala de testes a ser executada
- Plano de Testes
  - são construídos com base nos diversos modelos utilizados nas fases de especificação, análise, design e implementação
- Estratégia Geral de Testes
  - que tipos de testes serão executados, como executá-los, quando executá-los e como avaliar os resultados dos testes
  - testes custam recursos e tempo, portanto deve-se efetuar uma solução de testes que tenha uma boa relação custo/benefício

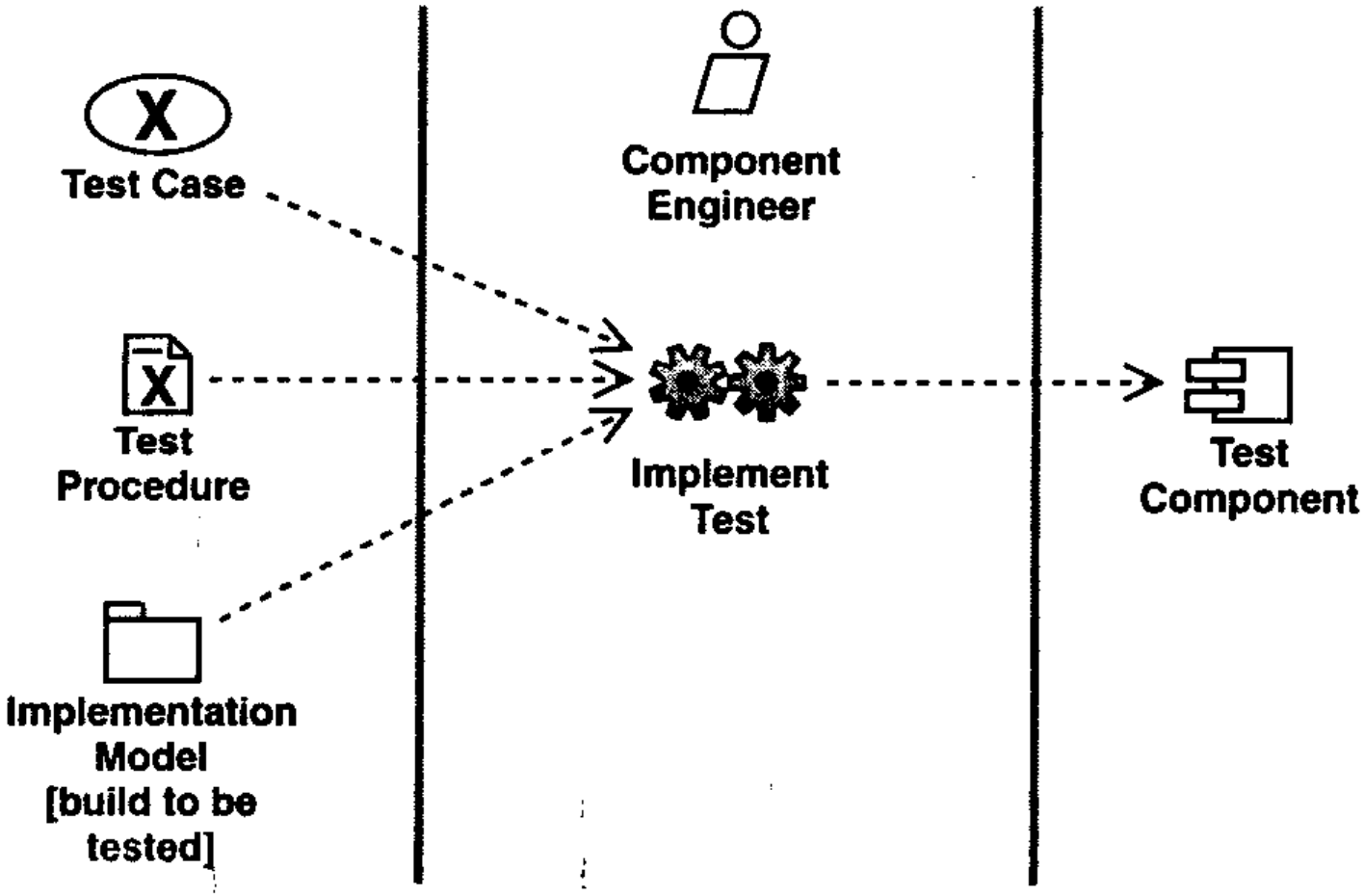
Projeto de Testes



Projeto de Testes

- Objetivos
  - identificar e descrever casos de teste para cada módulo
  - identificar e estruturar procedimentos de teste especificando como realizar os casos de teste
- Sub-Etapas
  - Projetando casos de teste de integração
  - Projetando casos de teste de sistema
  - Projetando casos de teste regressivos
    - aproveitam casos de teste de iterações anteriores
  - Identificando e Estruturando Procedimentos de Teste
- Regra Geral
  - deve-se utilizar um conjunto de testes que requeiram um mínimo esforço, dado um plano de testes
    - mínimo "overlap" entre casos de teste

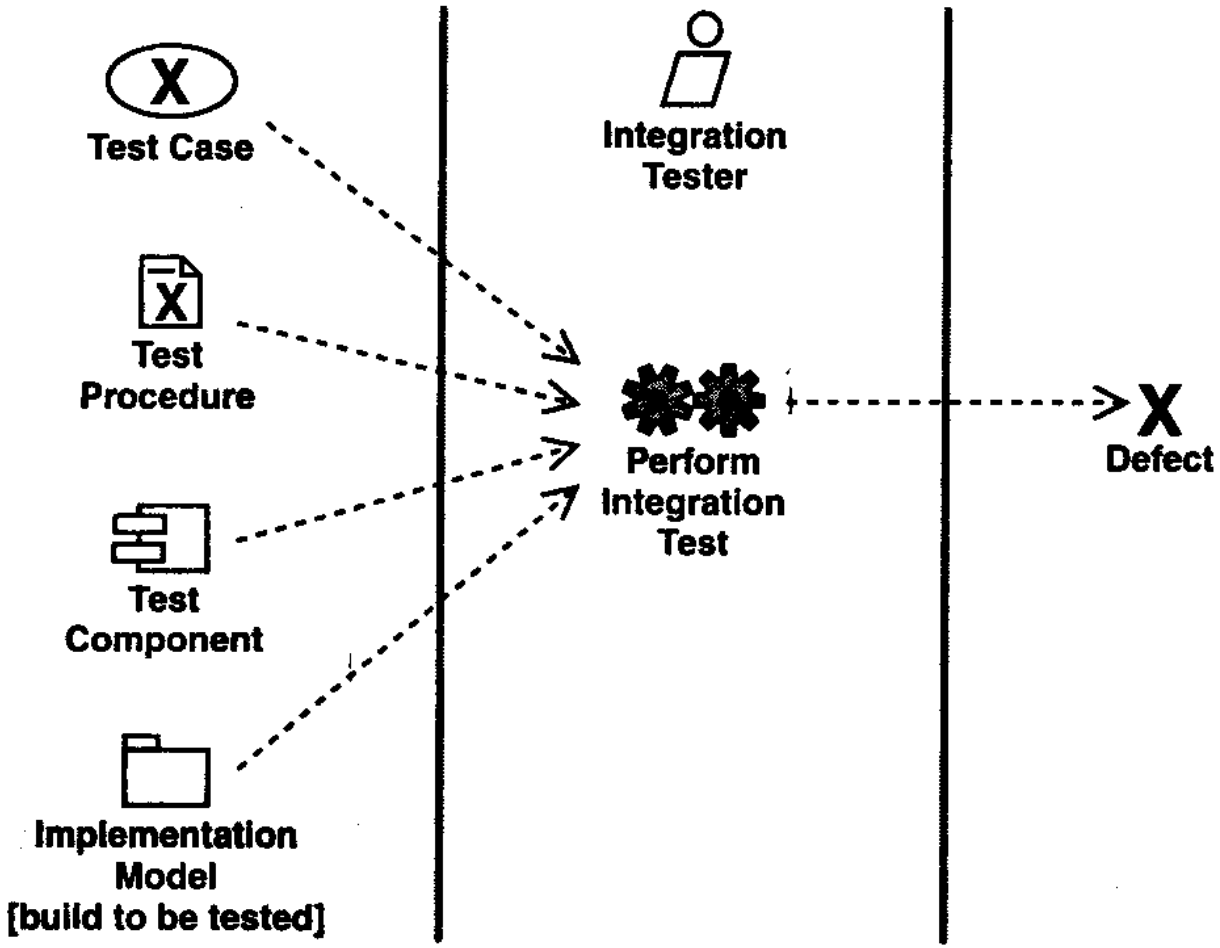
Implementação de Testes



Implementação de Testes

- Objetivos
  - automatizar procedimentos de teste por meio da criação de componentes de teste, se possível (nem todos os procedimentos de teste podem ser automatizados)
- Componentes de Teste
  - criados utilizando os procedimentos de teste como entrada
  - quando se utiliza uma ferramenta de automação, um conjunto de ações são previamente criadas para que o módulo em questão seja testado. A própria ferramenta se encarrega de criar os componentes de teste
  - quando programando os componentes de teste explicitamente, utiliza-se os procedimentos de teste como uma forma de especificação para que os componentes de teste sejam desenvolvidos

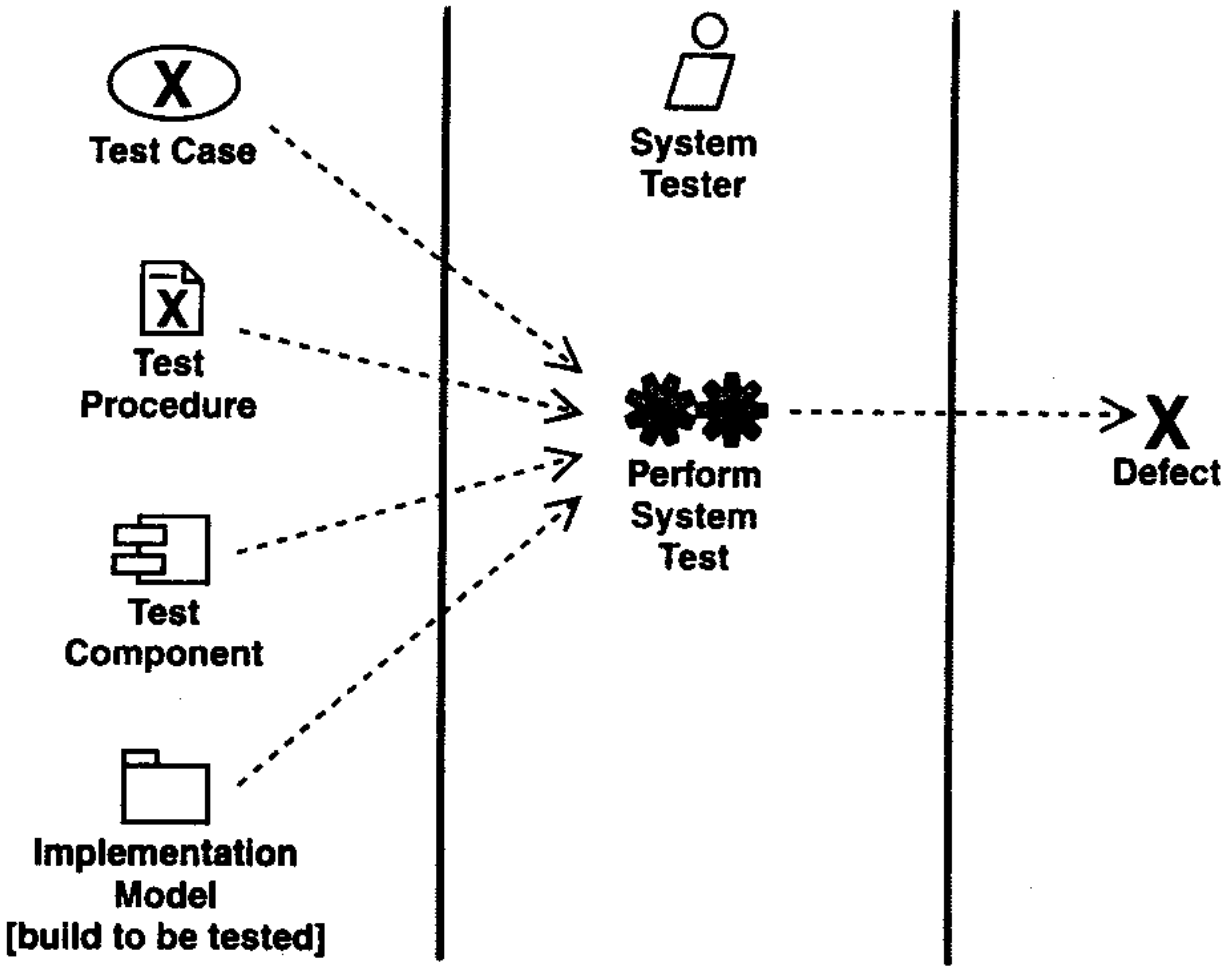
Teste de Integração



Teste de Integração

- Objetivos
  - realizar os testes de integração especificados para cada módulo do sistema
- Sequência de Testes de Integração
  - realizar os testes de integração relevantes por meio da execução manual dos procedimentos de teste para cada caso de teste ou por meio de algum componente de teste disponível para o procedimento de teste em questão
  - comparação dos resultados com os resultados esperados e a discriminação dos casos que apresentaram resultados inesperados
  - relatar os defeitos encontrados ao engenheiro de componentes, para que estes possam corrigir os componentes defeituosos
  - relatar os defeitos ao engenheiro de testes, para que este possa estimar os resultados finais da sequência de testes

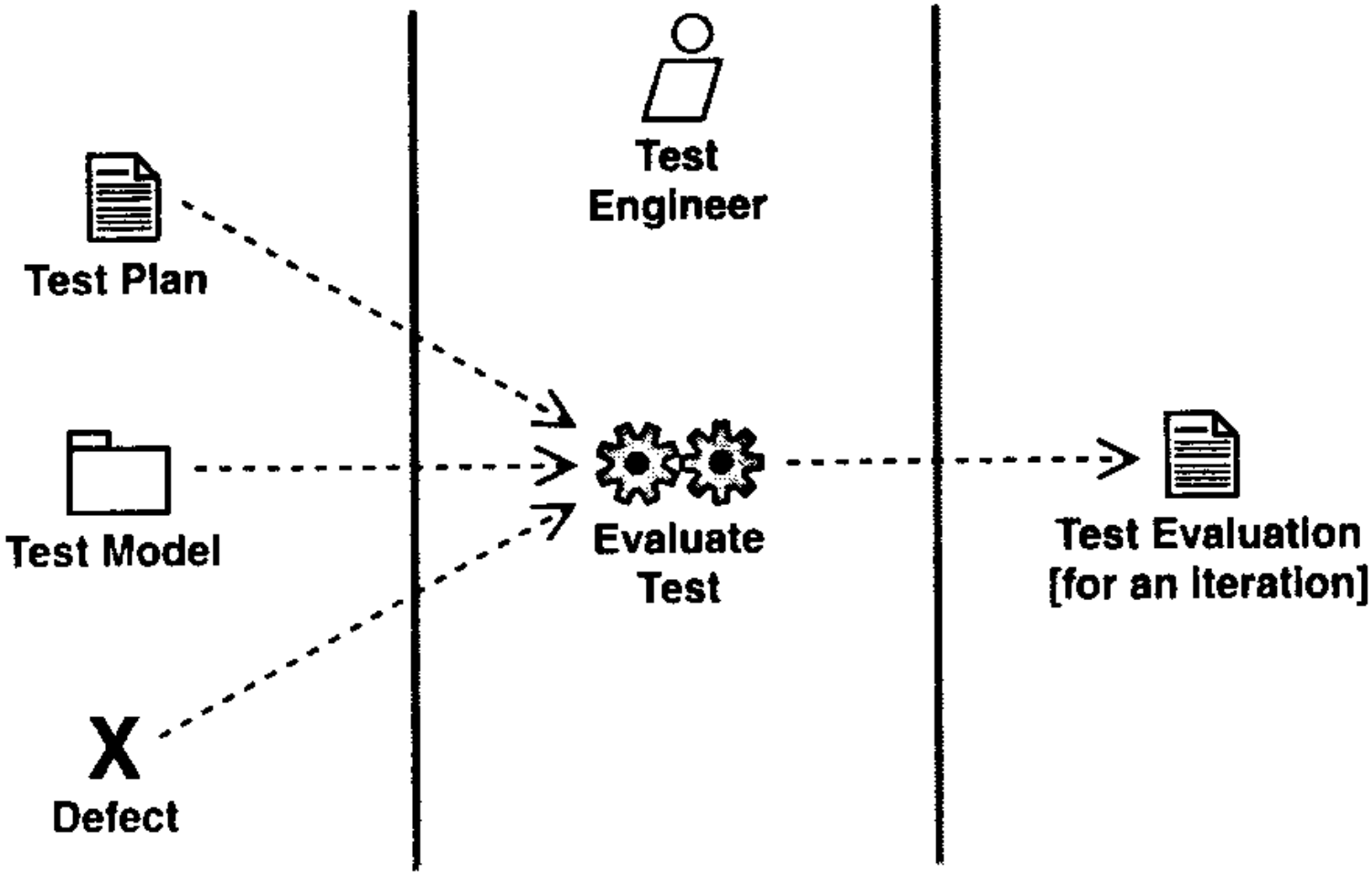
Teste de Sistema



Teste de Sistema

- Objetivos
  - realizar os testes de sistema especificados a cada iteração, capturando os resultados do teste
- Teste de Sistema
  - se inicia quando os testes de integração indicam que o sistema atende as metas de qualidade quanto a integração de seus componentes para a iteração corrente (e.g. 95 % dos testes de integração executam com resultado esperado)
  - são realizados de maneira análoga aos testes de integração (i.e. seguindo uma sequência de testes análoga ao dos testes de integração
  - avaliam o comportamento do funcionamento do sistema como um todo, e não somente com relação à integração individual entre alguns de seus componentes

Avaliação de Testes



Avaliação de Testes

- Objetivo
  - avaliação dos esforços de teste para a iteração corrente
    - comparação dos resultados com as metas especificadas no planejamento de testes
    - criação de métricas que determinem o nível de qualidade do software, especificando maiores testes que necessitem ser realizados
- Exemplos de Métricas
  - Métricas de Completude
    - derivadas da cobertura dos casos de teste e dos componentes de teste. Indicam qual a porcentagem de casos de teste que foram executados e qual a porcentagem de código que foi testada
  - Métricas de Confiabilidade
    - baseadas na análise dos defeitos descobertos com relação aos casos que tiveram um resultado como esperado