

Fase de Construção (*Constrution*)

OpenUP

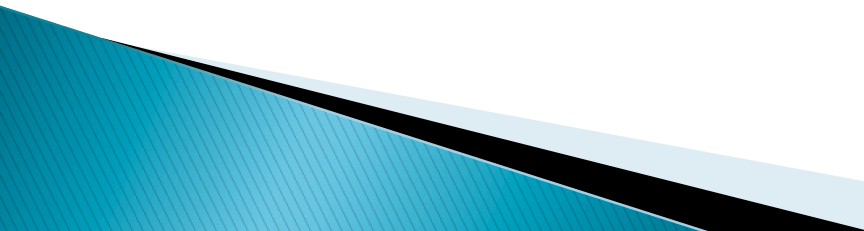
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Diretoria de Educação e Tecnologia da Informação
Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

Prof. Fellipe Aleixo (fellipe.aleixo@ifrn.edu.br)

Fase de Construção

- ▶ A arquitetura deve estar **estável**
 - Permitindo que os requisitos restantes sejam implementados sobre ela
- ▶ Outra vantagem da validação da arquitetura e eliminação da maior parte dos riscos é que a fase de construção fica mais **previsível**
 - Permite ao gerente de projeto focar na **eficiência da equipe** e **redução de custos**

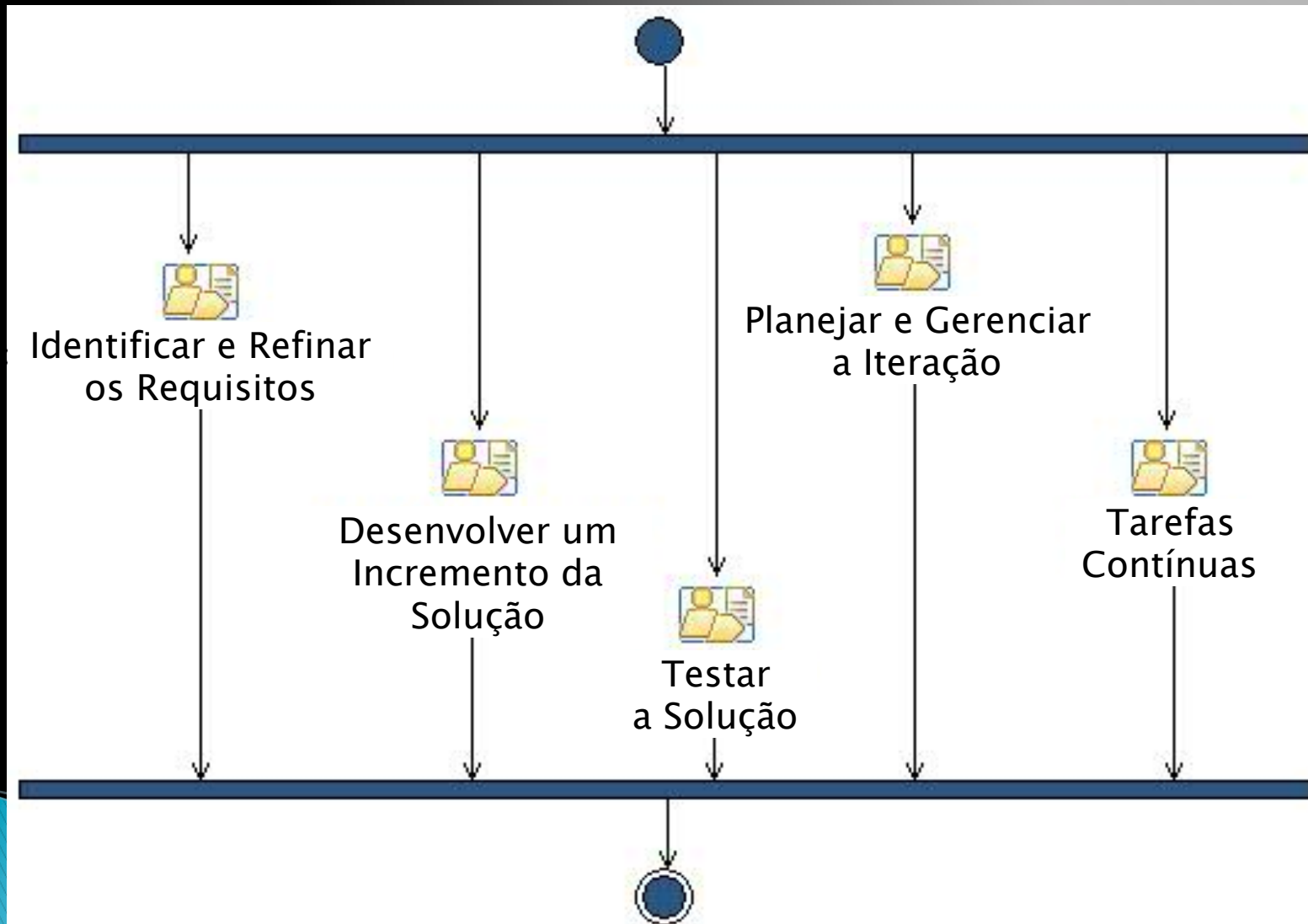
Fase de Construção

- ▶ As funcionalidades do sistema serão continuamente:
 1. Implementadas
 2. Testadas
 3. Integradas
 - ▶ Devem ser gerados *builds* cada vez mais completos e estáveis
 - ▶ Próximo ao final da fase é importante que seja realizada a implantação de um “beta” ou “*prerelease*” (antecipando a próxima fase)
- 

Objetivos e Atividades

Objetivos da Fase	Atividades que Endereçam os Objetivos
<p>Desenvolver iterativamente um produto completo, que esteja pronto para sofrer a transição para o ambiente do usuário</p>	<ul style="list-style-type: none">(i) Identificar e refinar os requisitos(ii) Desenvolver um incremento da solução(iii) Testar a solução
<p>Minimizar os custos de desenvolvimento e alcançar um certo grau de paralelismo</p>	<ul style="list-style-type: none">(i) Planejar e gerenciar a iteração(ii) Desenvolver um incremento da solução(iii) Testar a solução

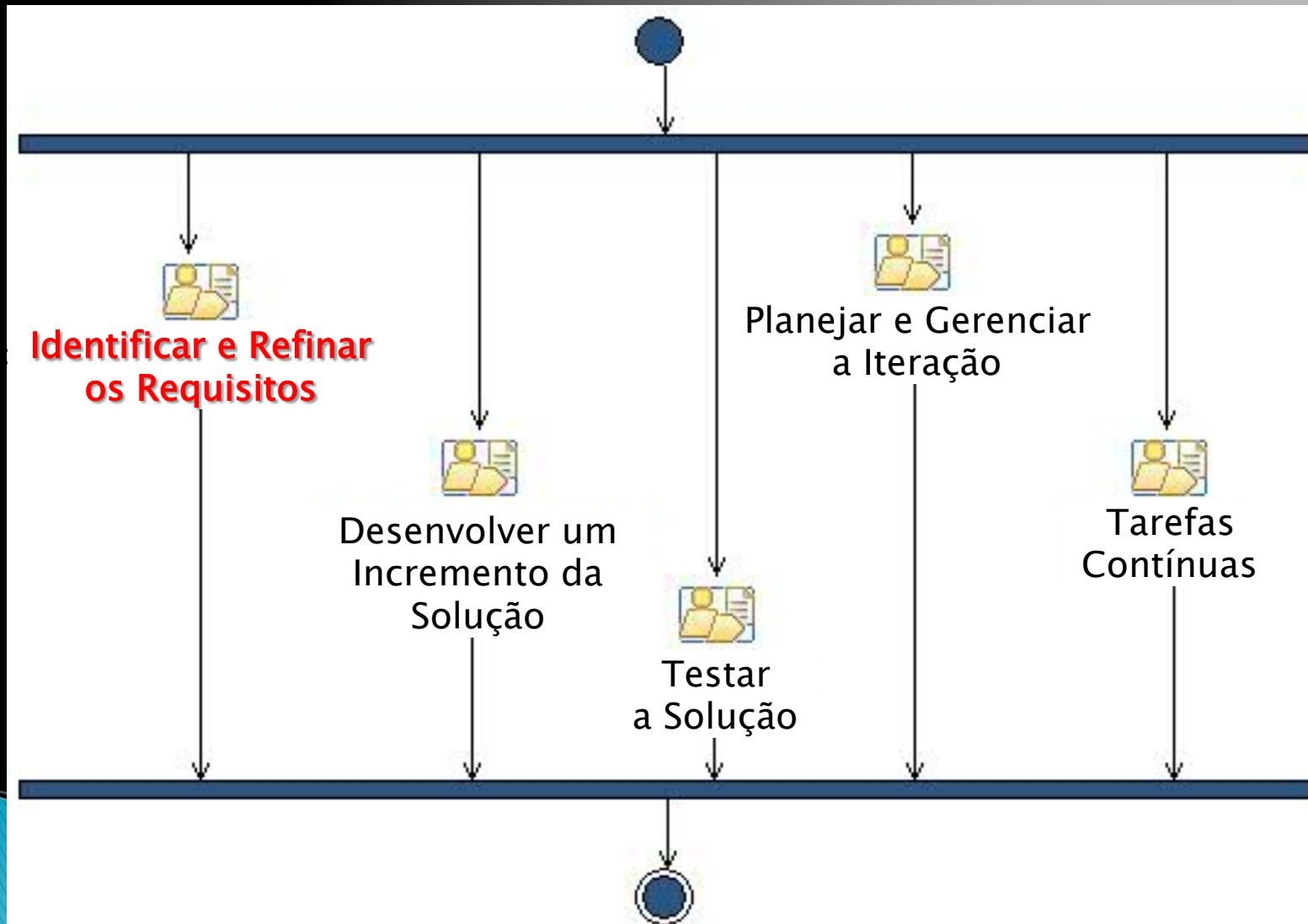
Estrutura de Divisão de Trabalho



Atividades

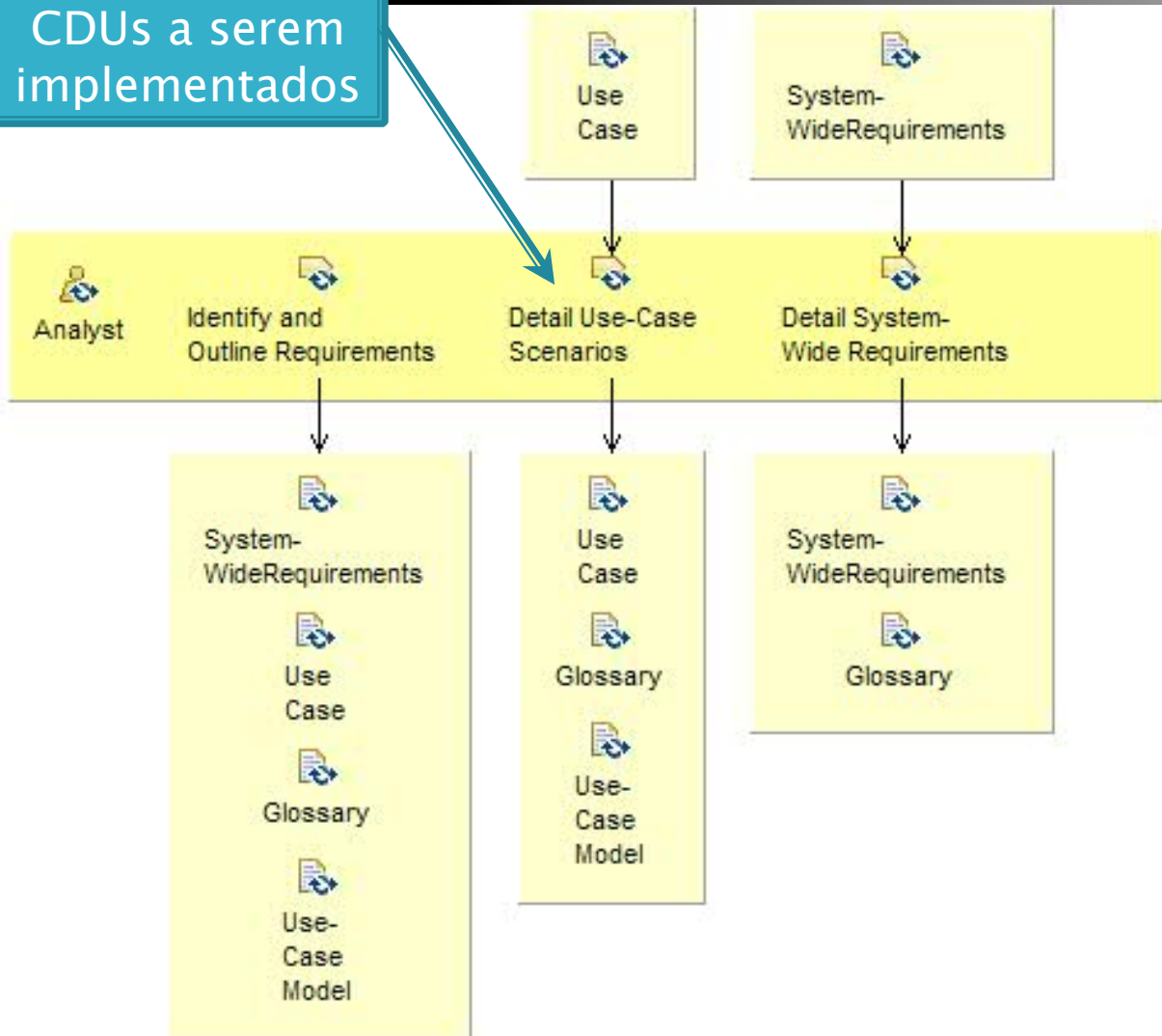
- » Descrição das atividades, suas tarefas e os respectivos passos

Atividades a Serem Executadas

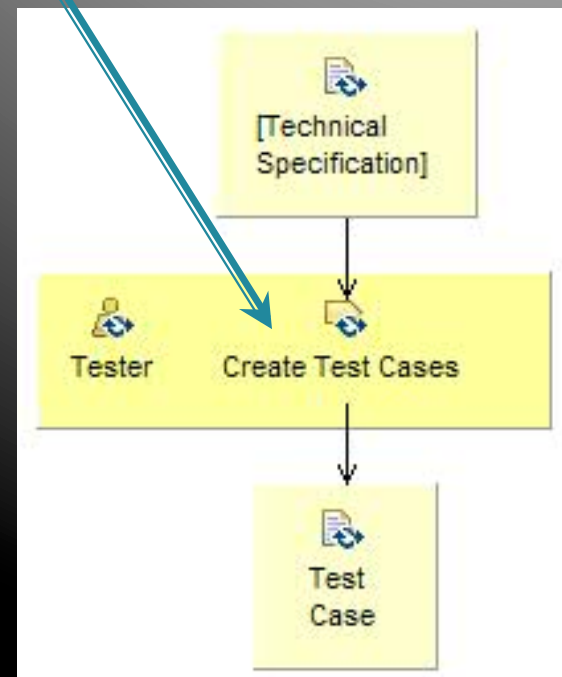


Identificar e Refinar os Requisitos

Detalhar os CDUs a serem implementados



Verificar se a implementação corresponde aos requisitos



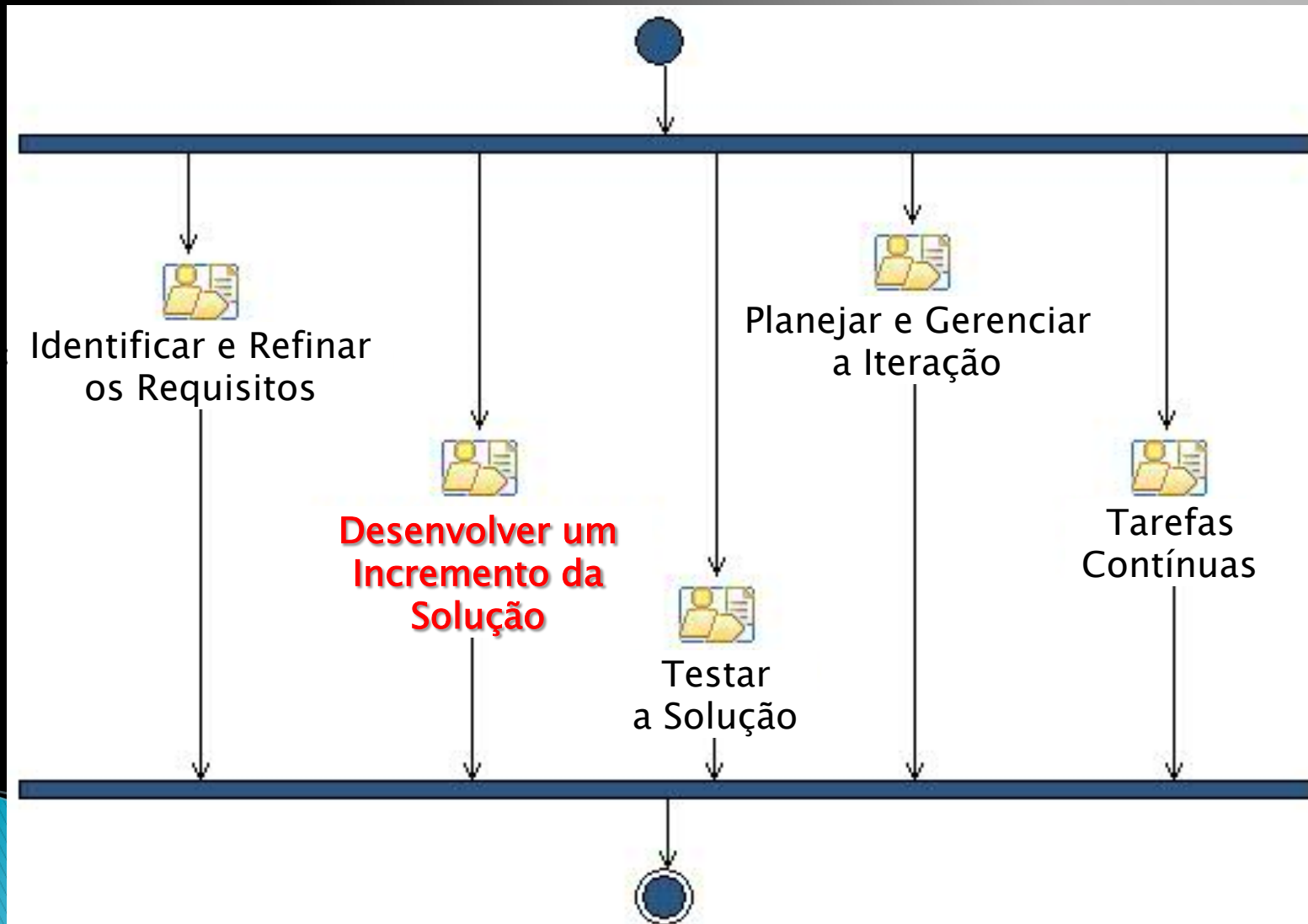
Tarefa: Detalhar os Cenários de Casos de Uso

- ▶ Descrever os cenários detalhados para os casos de uso do sistema
 - Com o **nível de detalhes suficiente para validar o entendimento dos requisitos**, e garantir a concordância com as expectativas dos *stakeholders*
- ▶ Passos:
 1. Detalhar os casos de uso e cenários
 2. Atualizar o modelo de casos de uso
 3. Detalhar os termos do glossário
 4. Atingir a concordância de todos

Tarefa: Criar os Casos de Teste

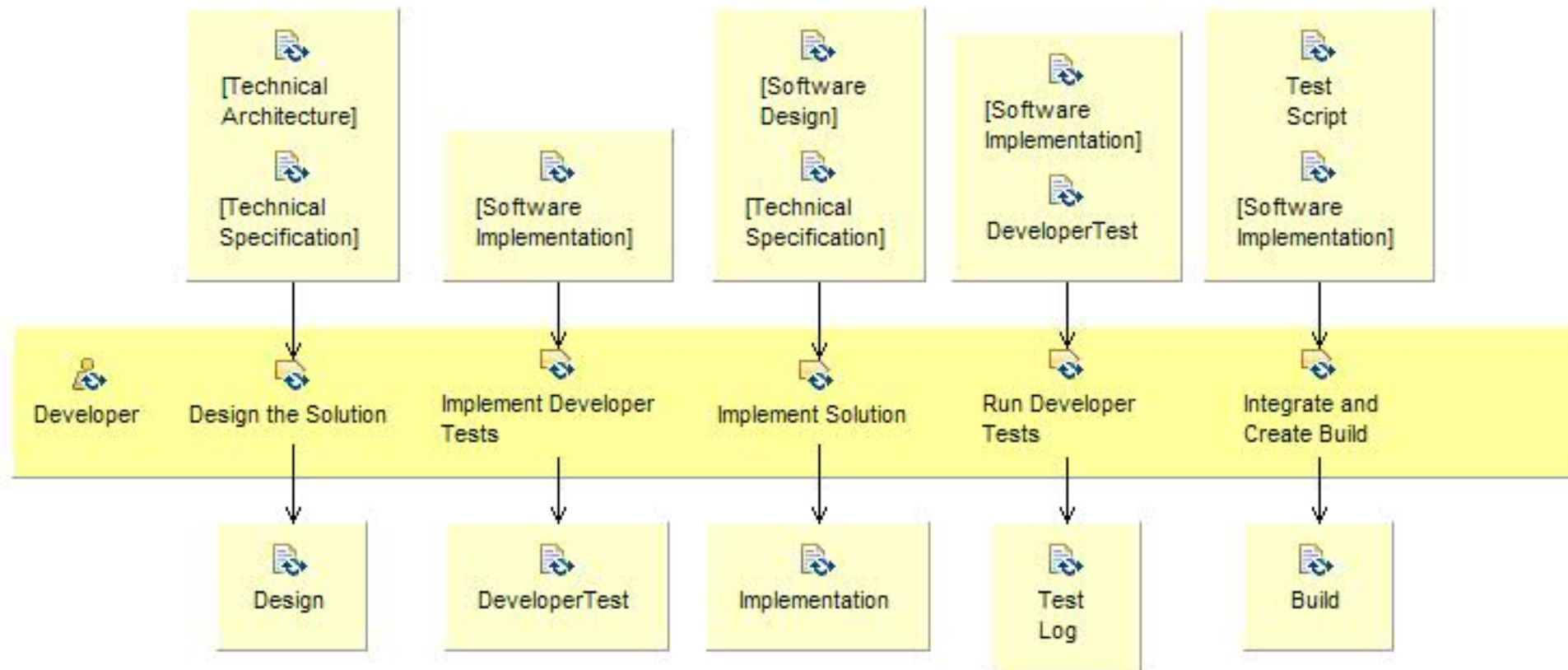
- ▶ Desenvolver os casos de teste e dados de teste para os requisitos a serem testados
 - Atingir um entendimento compartilhado sobre as **condições específicas que o sistema deve atender**
- ▶ Passos:
 1. **Revisar os requisitos a serem testados**
 2. **Identificar casos de teste relevantes**
 3. Esboçar os casos de teste
 4. Identificar os dados de teste necessários
 5. Compartilhar e avaliar os casos de teste

Atividades a Serem Executadas

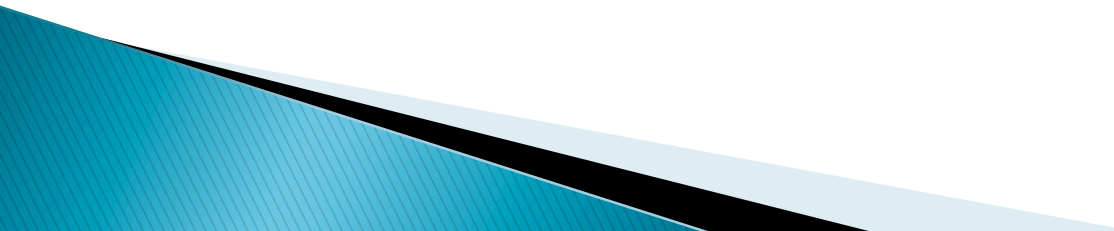


Desenvolver um Incremento da Solução

- ▶ Devem ser estudadas estratégias de paralelismo das ações de implementação



Ordem das Tarefas

1. Projetar a solução
 - Diagramas de interação, [de estados], [de atividades], e de classes de projeto
 2. Implementar os testes de desenvolvedor
 - Implementar os teste unitários (teste de classe)
 3. Implementar a solução
 - Codificar segundo os padrões de codificação
 4. Executar os testes de desenvolvedor
 - Executar os testes unitários, corrigindo os erros
 5. Integrar e gerar um *build*
- 

Visão Geral do Fluxo de Trabalho

- ▶ **Não é necessário** fazer o projeto para todos os casos de uso sendo desenvolvidos
- ▶ O projeto específico para alguns casos de uso podem ser **evidentes** no contexto da arquitetura e dos projetos de outros casos de uso já implementados
- ▶ Se um caso de uso específico abordar aspectos ainda não implementados, pode ser necessário um **esforço em projetar a solução** antes de iniciar a implementação
 - Mesmo que seja apenas um exercício mental

Visão Geral do Fluxo de Trabalho

- ▶ Estando clara a organização da solução técnica é importante definir os **testes de desenvolvedor**, que irão verificar a implementação
- ▶ A **abordagem orientada a testes** garante que considerações de projeto sejam de fato observadas antes que a solução seja codificada
- ▶ A cada avanço na implementação são executados os testes de desenvolvedor (**testes unitários**)
 - Estes definem claramente o **critério** que determina se a implementação está de acordo com o esperado

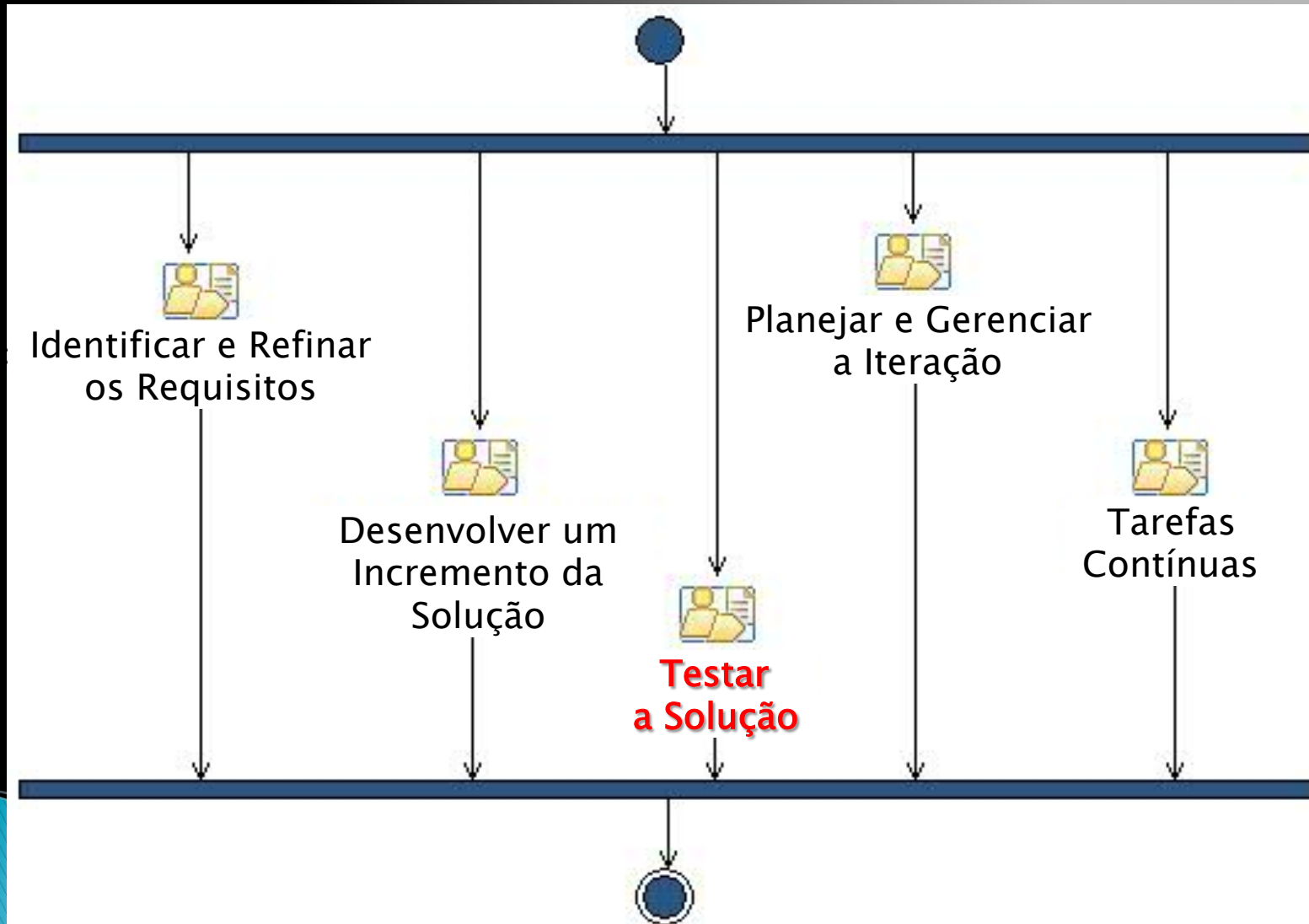
Visão Geral do Fluxo de Trabalho

- ▶ A cada pequeno passo (p.ex.: implementação de um método) os testes são executados
 - Se o teste específico falha, voltamos para a implementação da solução, testando novamente quando concluir a implementação
 - Esse laço de repetição é executado até que o teste específico seja bem sucedido
 - Se a implementação do caso de uso ainda não estiver completa é selecionado um novo pequeno avanço para ser implementado

Visão Geral do Fluxo de Trabalho

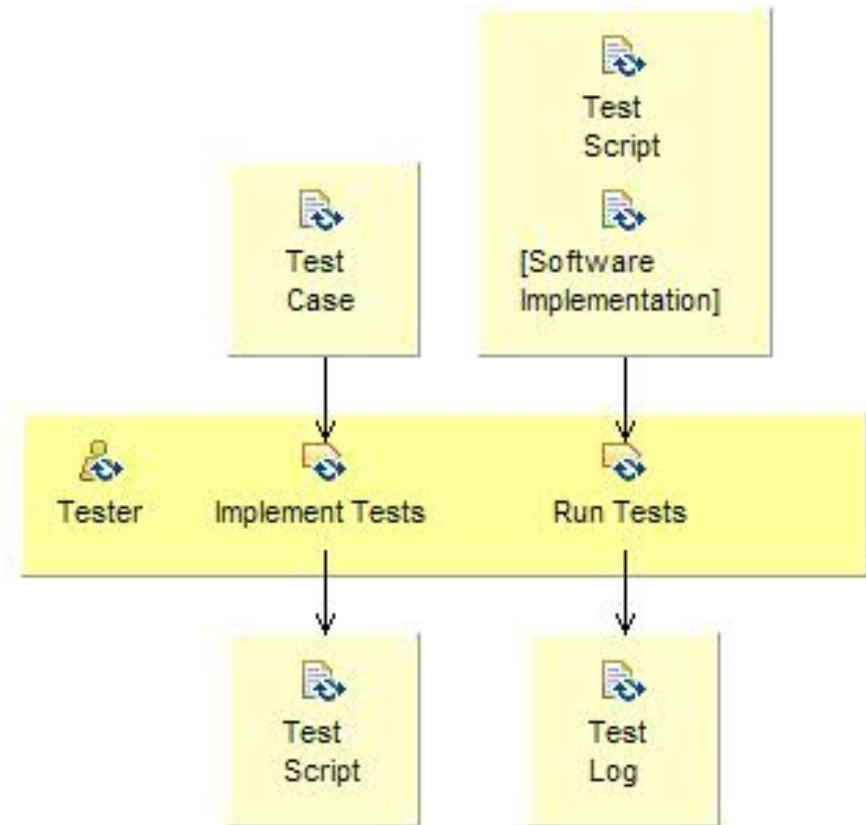
- ▶ Passar nos testes não implica que a solução seja necessariamente apropriada e de alta qualidade
 - Para proceder com esse tipo de análise, devemos **revisitar o projeto da solução**
- ▶ E em caso de qualquer alteração no projeto, todo o processo deve ser repetido, dado que qualquer mudança no projeto pode afetar os testes de desenvolvedor e a implementação

Atividades a Serem Executadas



Testar a Solução

- ▶ Desenvolver e executar testes que garantem que os casos de uso selecionados foram implementados conforme especificados



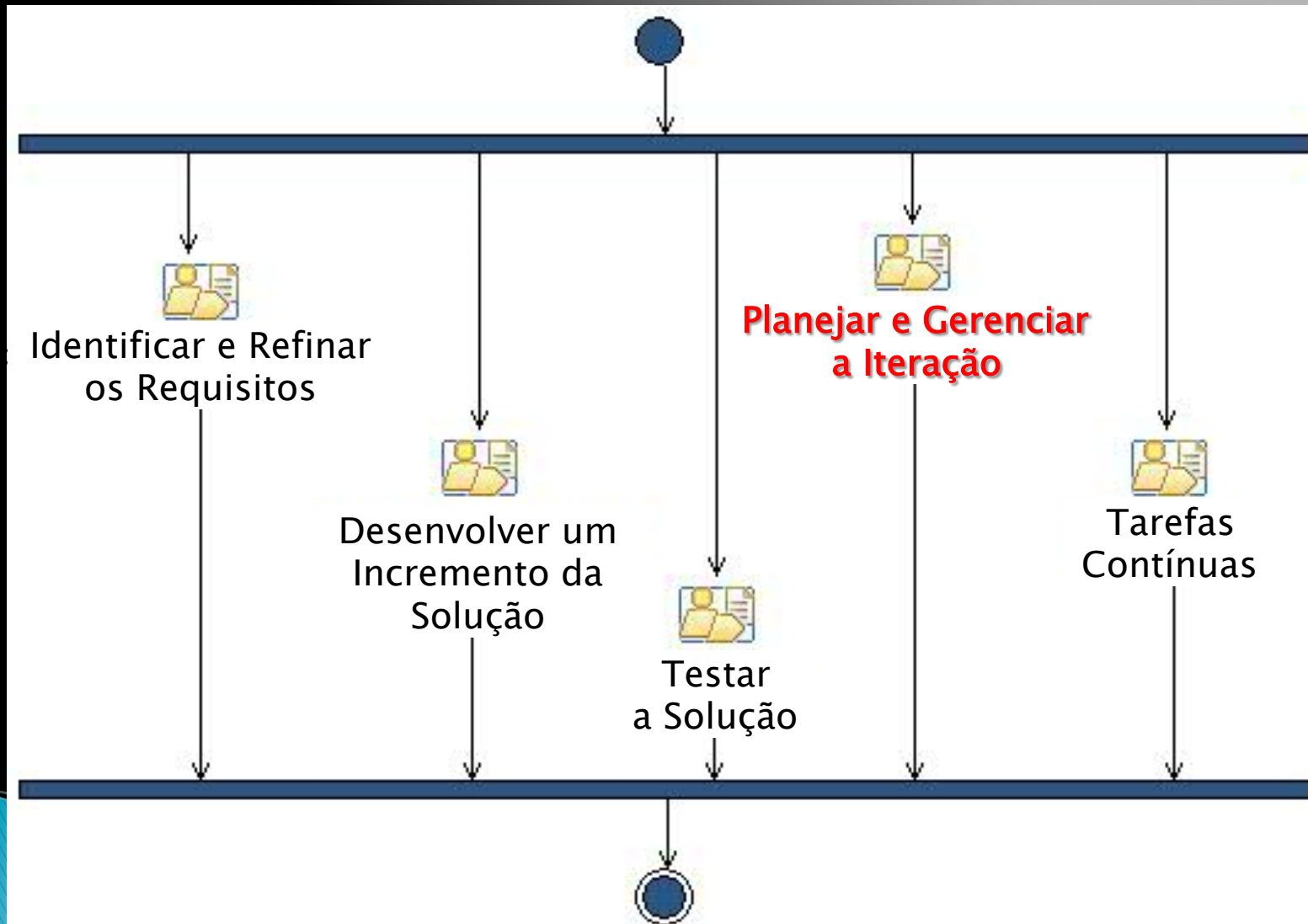
Tarefa: Implementar os Teste

- ▶ Implementar roteiros de teste passo-a-passo que **demonstrem que a solução satisfaz os requisitos**
- ▶ Passos:
 1. Selecionar os casos de teste a serem implementados
 2. Projetar o roteiro de testes
 3. Implementar um **roteiro de testes executável**
 4. Definir dados de teste específicos
 5. Organizar os roteiros de teste em “suítes”
 6. Verificar a implementação dos testes
 7. Compartilhar e avaliar os roteiros de teste

Tarefa: Executar os Testes

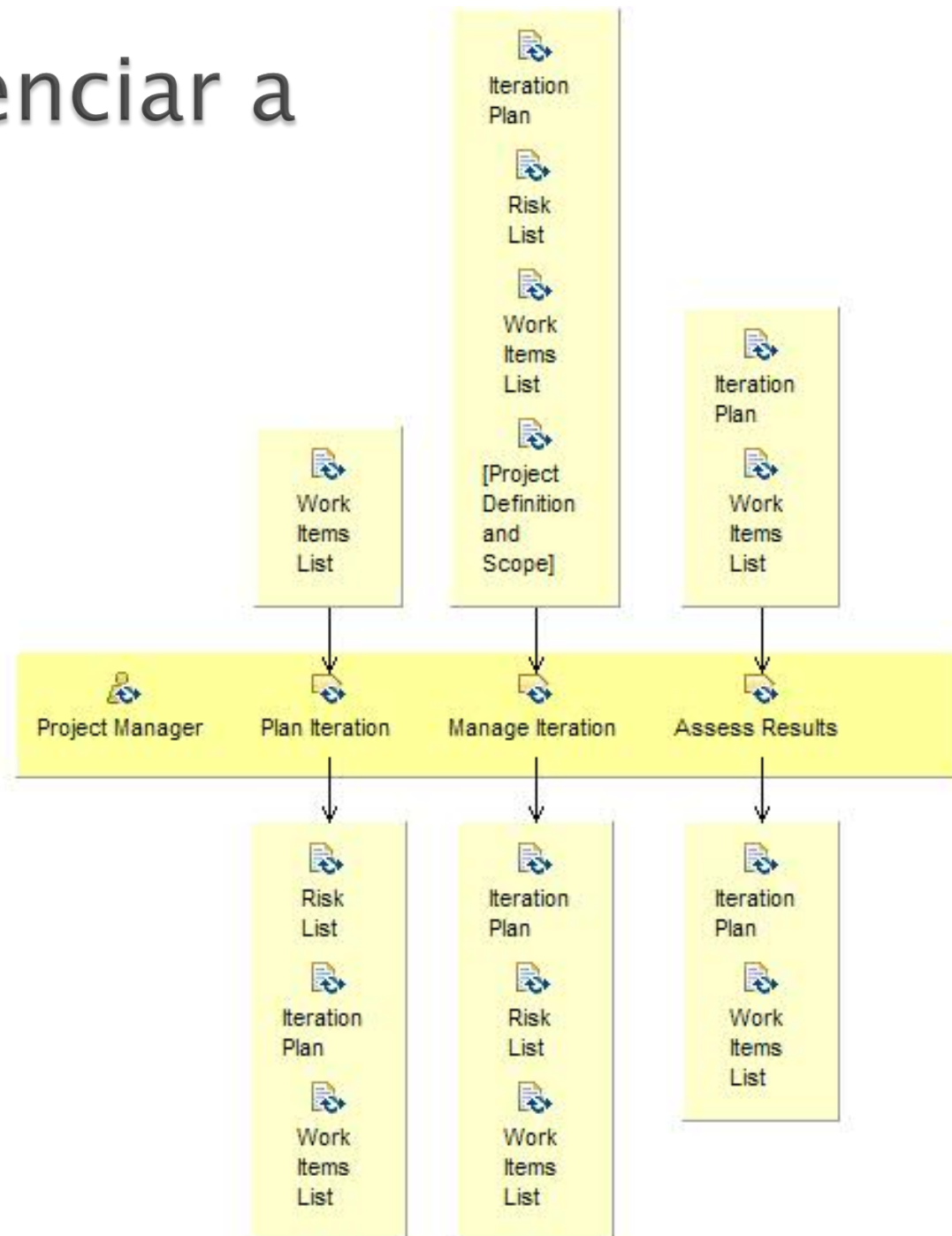
- ▶ Possibilitar um feedback para a equipe de quão bem o *build* satisfaz os requisitos selecionados
- ▶ Passos:
 1. Rever os itens de trabalho completados no *build*
 2. **Selecionar** os roteiros de testes
 3. **Executar** os roteiros de testes no *build*
 4. **Analisar e comunicar os resultados** dos testes
 5. Prover um feedback para a equipe

Atividades a Serem Executadas

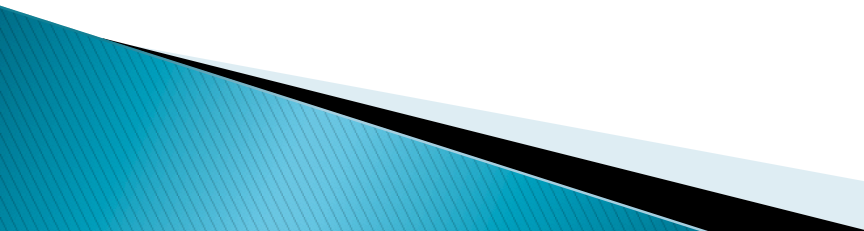


Planejar e Gerenciar a Iteração

- ▶ Distribuição das tarefas de desenvolvimento entre os membros da equipe
- ▶ Monitorar e informar o status para os *stakeholders*
- ▶ Identificar e gerenciar as exceções e problemas



Planejar e Gerenciar a Iteração

- ▶ Esta atividade é realizada durante todo o ciclo de vida do projeto
 - ▶ O objetivo principal dessa atividade é:
 - a **identificação antecipada de riscos e problemas**, para que os mesmos possam ser mitigados
 - **Estabelecer metas claras e compartilhadas para a iteração** (buscando o comprometimento de todos)
 - **Dar suporte à equipe** de desenvolvimento para que alcancem as metas definidas
- 

Planejar e Gerenciar a Iteração

- ▶ Orientações gerais para gerentes e equipe:
 1. Deve ser realizada uma **priorização do trabalho** atribuído para a iteração
 2. Todos devem **concordar com o que será desenvolvido** na iteração
 3. Um membro da equipe pode assumir a responsabilidade por “itens de trabalho”
 4. Um membro da equipe quebra um “item de trabalho” em tarefas de desenvolvimento e estima o seu esforço (estimação mais precisa e realista do tempo que se irá gastar)

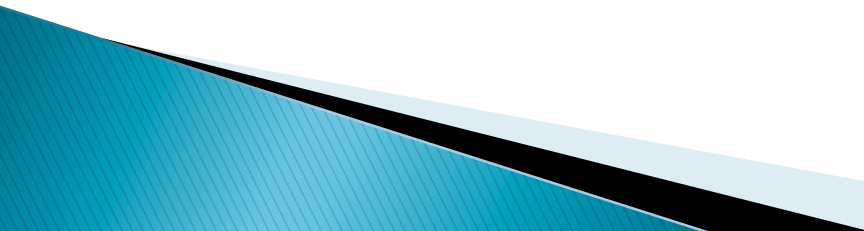
Planejar e Gerenciar a Iteração

- ▶ A medida que a iteração vai acontecendo, a equipe regularmente informa:
 1. O trabalho que **foi completado**
 2. O trabalho que **está sendo realizado**
 3. O próximo trabalho **a ser realizado**
 4. **Problemas que estão bloqueando o progresso**
- ▶ Essas informações podem ser passadas em uma **reunião diária** (rápida e objetiva)
- ▶ Em uma reunião diária a equipe pode discutir:
 - O *status* do trabalho
 - Apontar problemas e riscos (a serem atacados pelo gerente)
 - **Propor ajustes ou correções** para garantir que os objetivos da iteração sejam atingidos

Planejar e Gerenciar a Iteração

- ▶ Durante a avaliação da iteração, a chave para o sucesso é a demonstração que a **funcionalidade planejada foi devidamente implementada**
- ▶ Se o final da iteração coincidir com o final de uma fase, devem também ser observados os **objetivos da fase**
- ▶ **Lições aprendidas** devem ser devidamente discutidas e incorporadas ao processo, para o melhoramento do mesmo

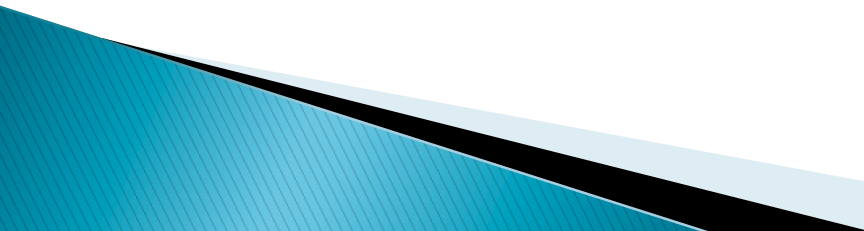
Tarefa: Planejar a Iteração

- ▶ Planejar o escopo e responsabilidades para uma dada iteração
 - ▶ Passos:
 1. Priorizar a lista de itens de trabalho
 2. **Definir os objetivos da iteração**
 3. Consignar trabalho à iteração
 4. Identificar e rever os riscos
 5. **Definir um critério de avaliação**
 6. Refinar a definição e escopo do projeto
- 

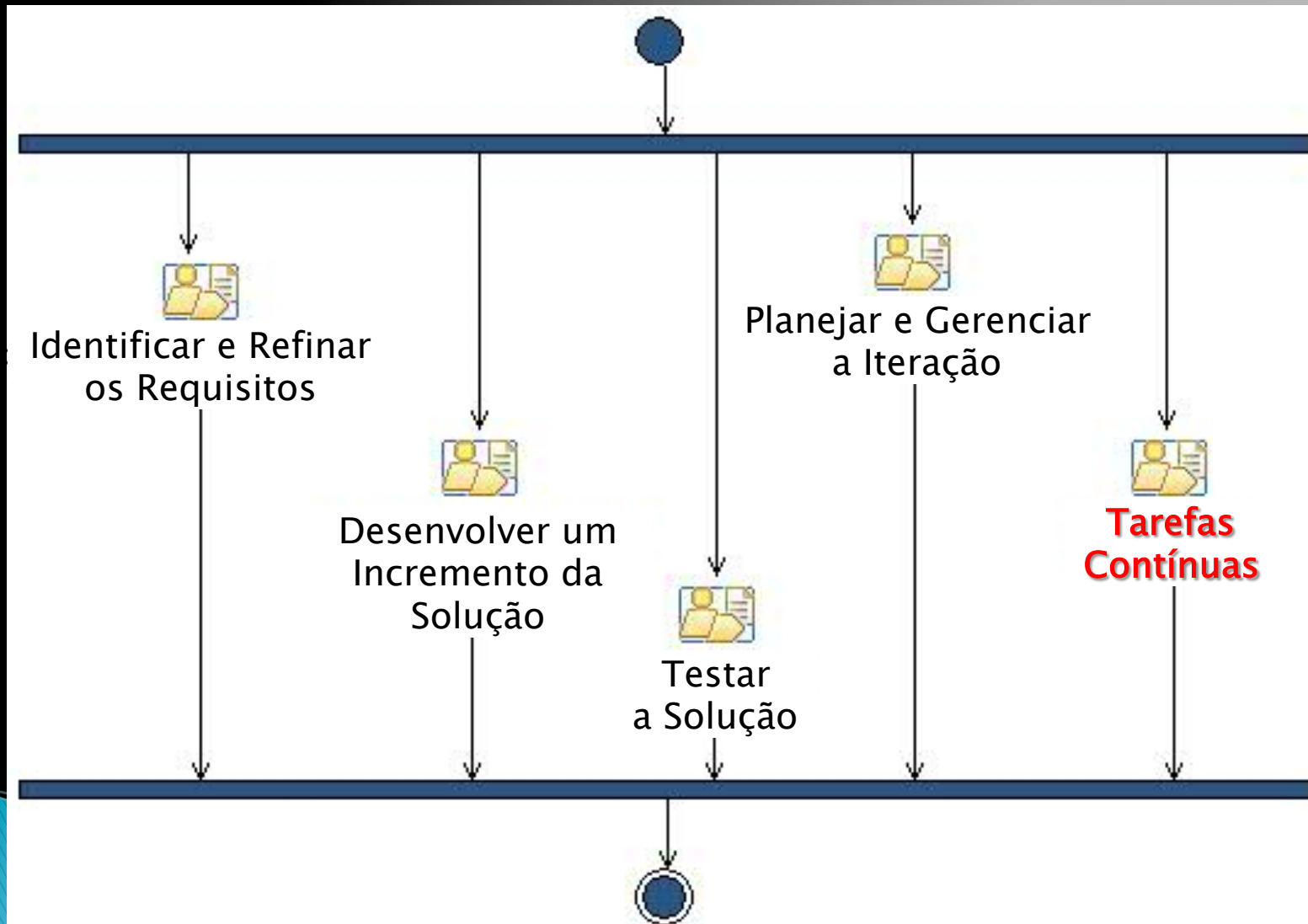
Tarefa: Gerenciar a Iteração

- ▶ Avaliar o status do projeto, identificando obstáculos e oportunidades
- ▶ Identificar e gerenciar as exceções, problemas e riscos
- ▶ Passos:
 1. Rastrear o progresso da iteração corrente
 2. Capturar e noticiar o status do projeto
 3. Gerenciar exceções e problemas
 4. Identificar e gerenciar os riscos
 5. Gerenciar os objetivos

Tarefa: Avaliar os Resultados

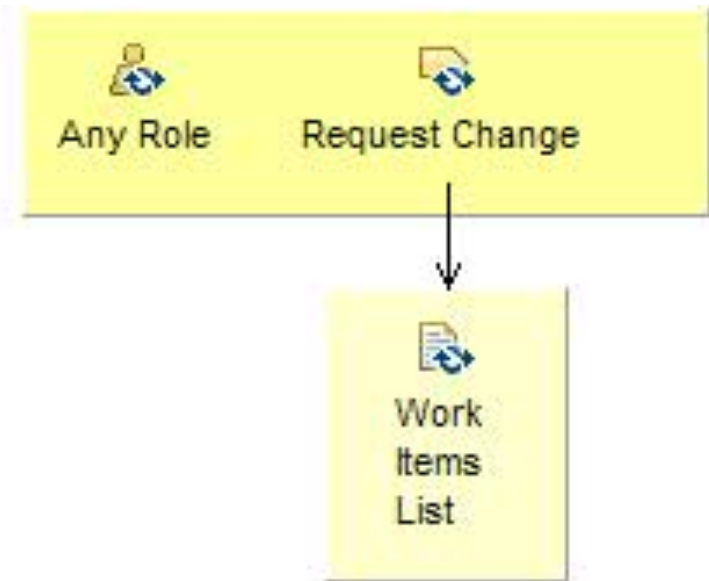
- ▶ Determinar o sucesso ou falha da iteração
 - ▶ Aplicar as lições aprendidas para modificar o projeto e aperfeiçoar o processo
 - ▶ Passos:
 1. Preparar para a avaliação da iteração
 2. **Demonstrar o valor e obter feedback**
 3. **Realizar uma retrospectiva**
 4. Realizar uma retrospectiva (final da fase)
 5. Reunião de finalização do projeto (última iteração)
- 

Atividades a Serem Executadas



Registrar Correções ou Mudanças

- ▶ Atividade que não faz parte do cronograma
- ▶ A tarefa de requisitar mudança pode ocorrer durante o ciclo de vida em resposta à observação de um **defeito**, um **melhoramento desejado**, ou uma **solicitação de mudança**



Tarefa: Solicitação de Mudança

- ▶ Qualquer membro da equipe pode ser responsável por registrar uma solicitação de mudança, os mais comuns são
 - *Stakeholders* – melhoramentos ou mudanças
 - Testadores – relato de defeitos
- ▶ Capturar e registrar requisições de mudança
- ▶ Passos:
 1. Obter informações sobre a solicitação de mudança
 2. Atualizar a lista de itens de trabalho