

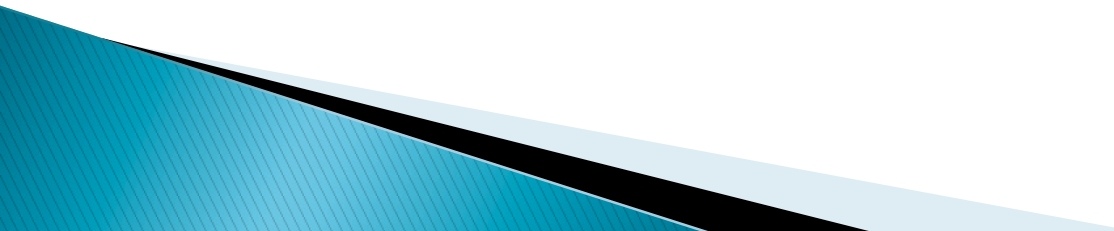
Fase de Elaboração (*Elaboration*)

OpenUP

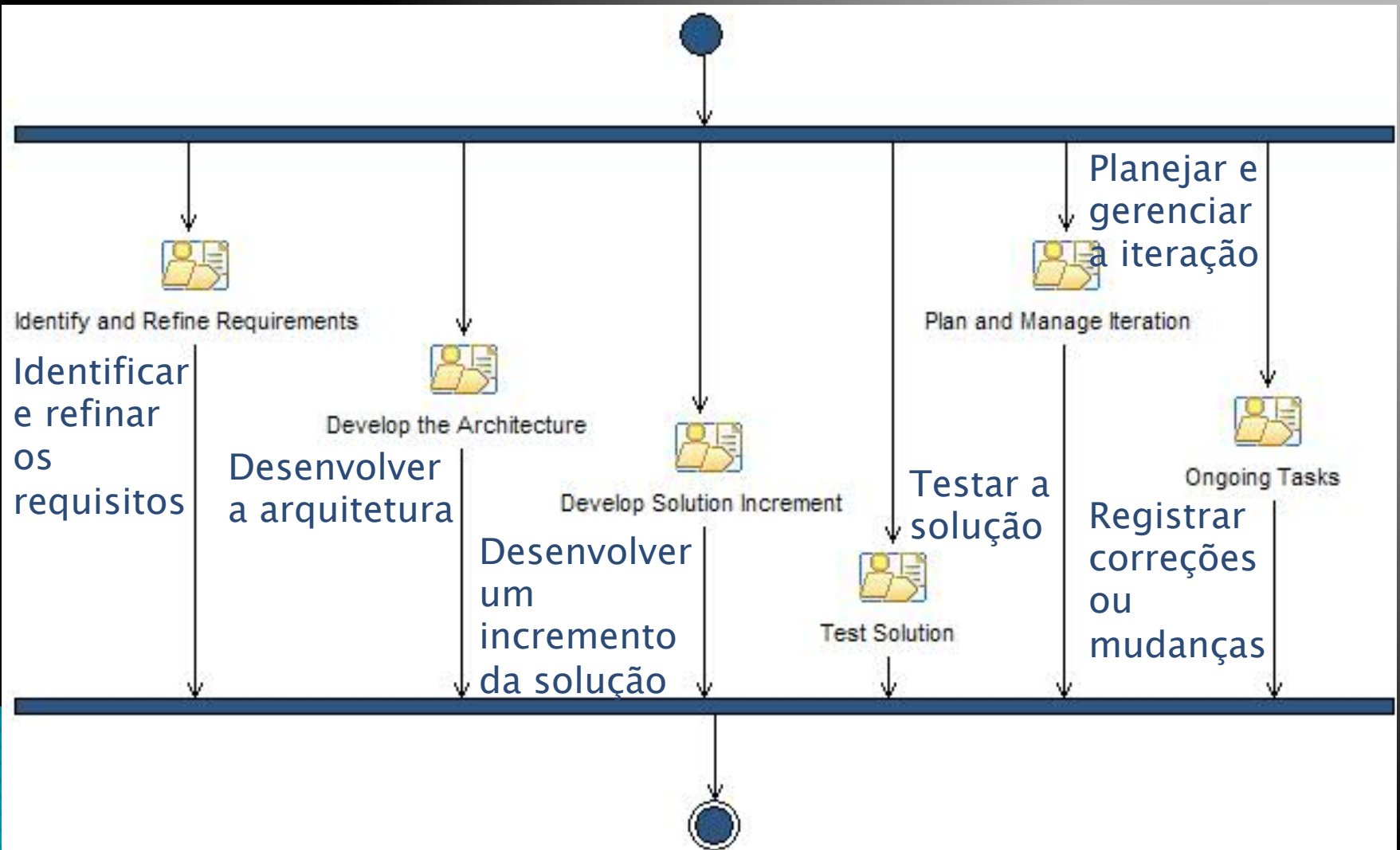
Instituto Federal de Educação, Ciência e Tecnologia do Rio Grande do Norte
Departamento Acadêmico de Gestão e Tecnologia da Informação
Curso de Tecnologia em Análise e Desenvolvimento de Sistemas

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

Objetivos da Fase de Elaboração

- ▶ Obter um entendimento mais detalhado dos requisitos do sistema
 - ▶ Projetar, implementar e validar a arquitetura de referência
 - ▶ Mitigar os riscos essenciais, e produzir um cronograma e uma estimativa de custos mais precisos
- 

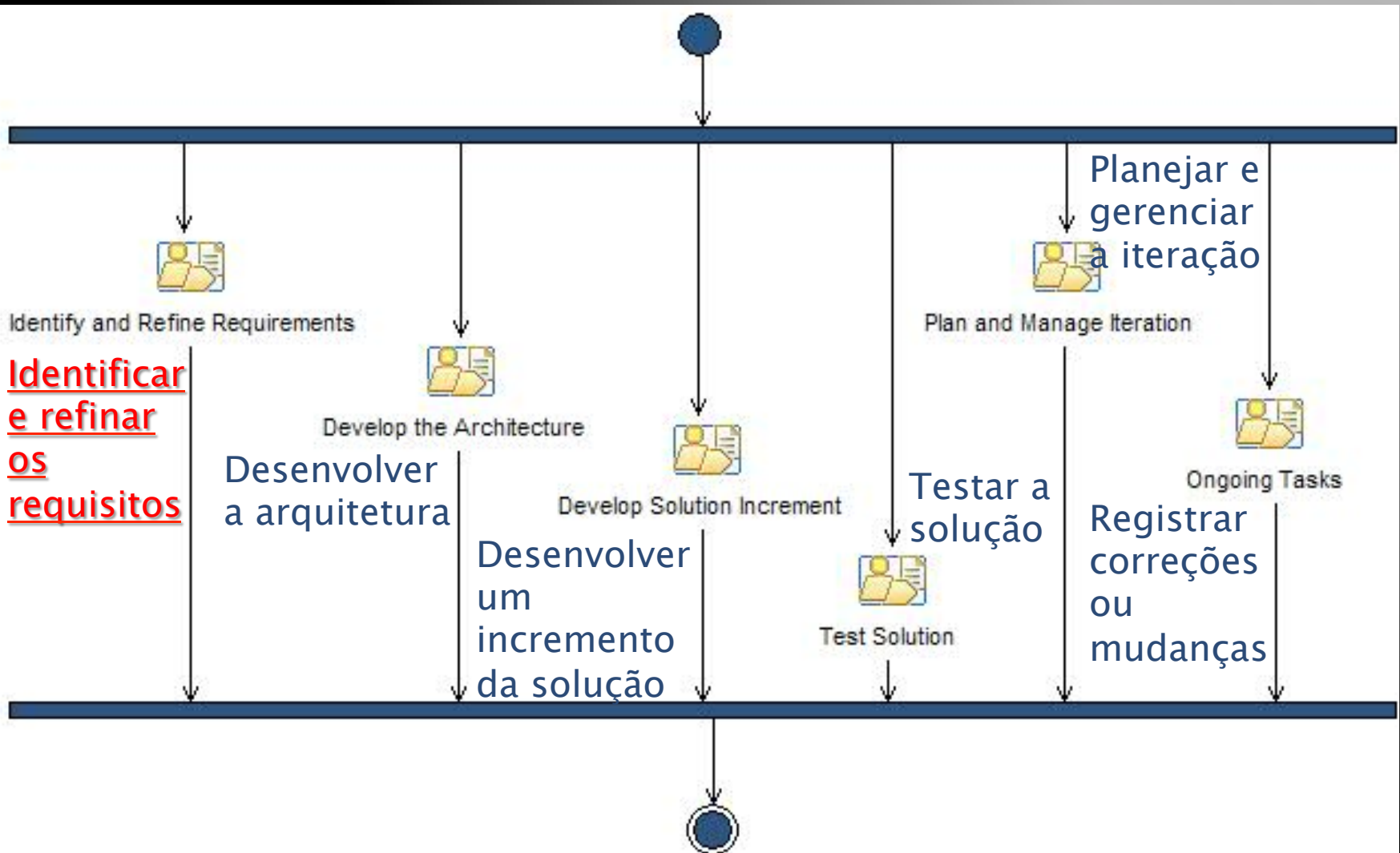
Atividades a Serem Executadas



Atividades

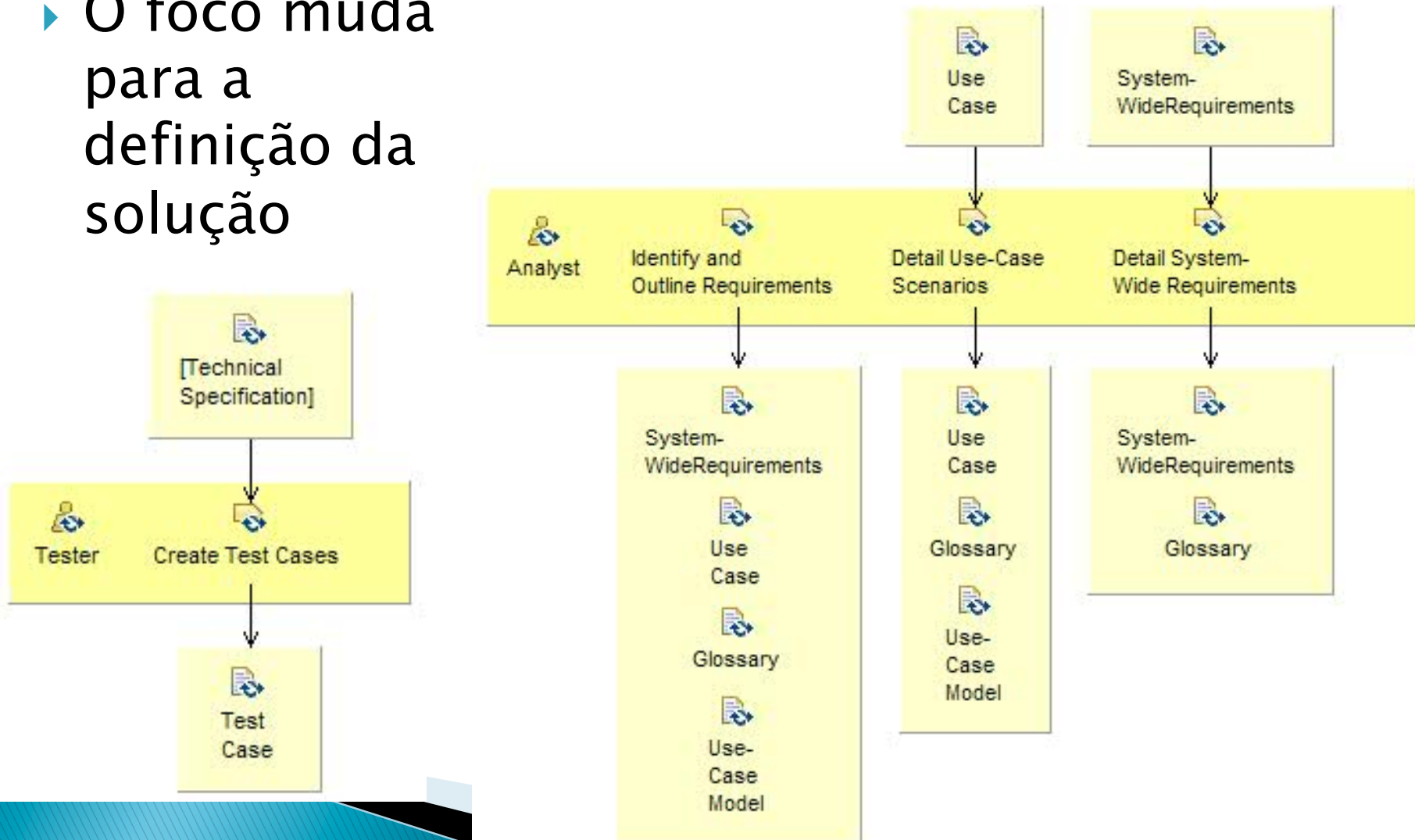
- » Descrição das atividades, suas tarefas e os respectivos passos

Atividades a Serem Executadas




Identificar e Refinar os Requisitos

- ▶ O foco muda para a definição da solução



Tarefa: Identificar e Esboçar os Requisitos

- ▶ Esboçar as os requisitos para o sistema, determinando o escopo do mesmo
 - ▶ Identificar e capturar requisitos funcionais e não-funcionais para o sistema
 - ▶ Passos:
 1. Agrupar as informações
 2. Identificar e capturar os termos de domínio
 3. Identificar os tipos de requisitos relevantes para o sistema
 4. Identificar e capturar casos de uso e seus atores
 5. Identificar e capturar requisitos do sistema como um todo
 6. Atingir a concordância de todos
- 

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

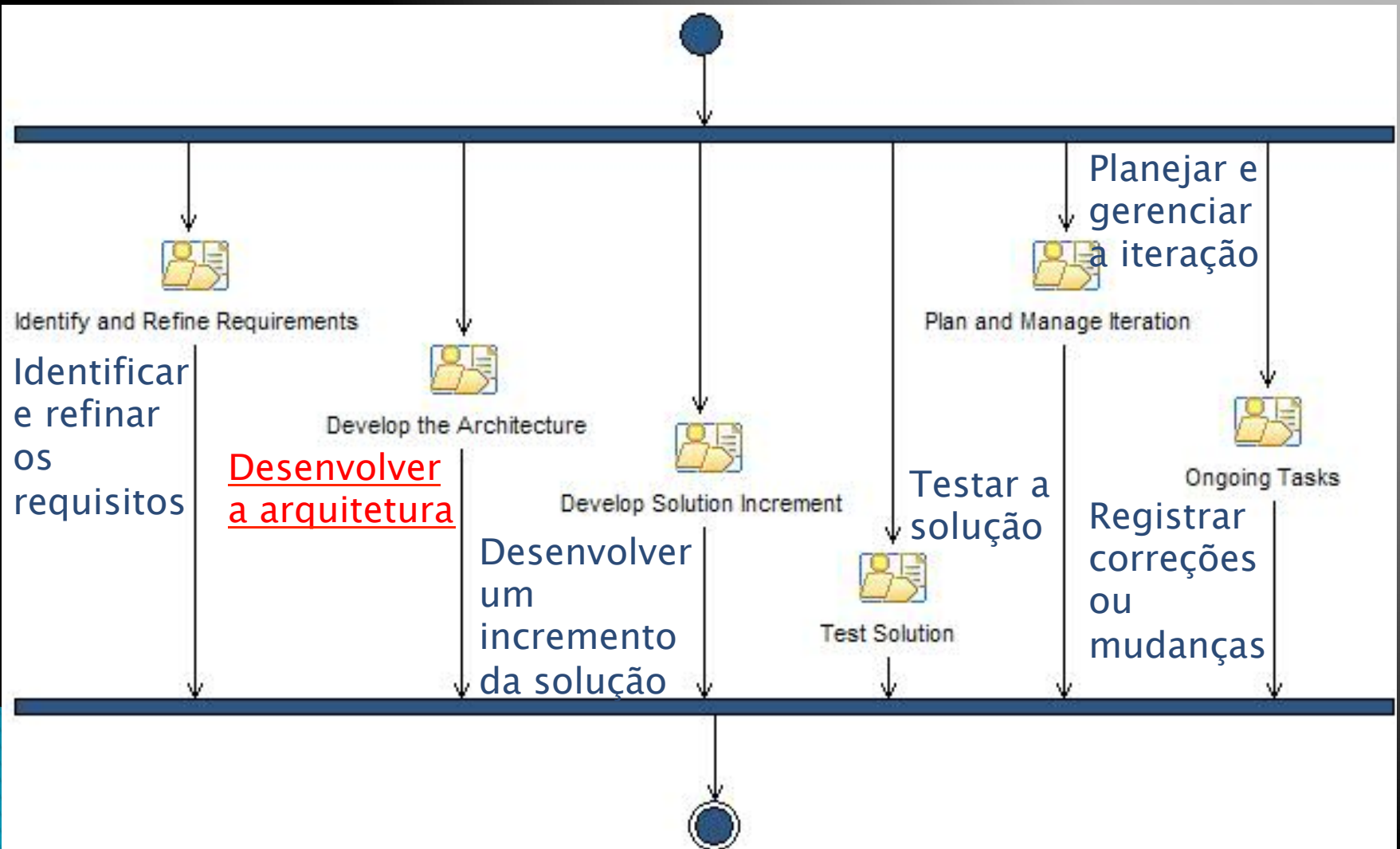
Tarefa: Detalhar os Requisitos do Sistema como um Todo

- ▶ Detalhar um ou mais requisitos que não se aplicam a nenhum caso de uso específico
 - 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 requisitos do sistema como um todo
 2. Detalhar os termos do glossário
 3. Atingir a concordância de todos

Tarefa: Criar os Casos de Teste

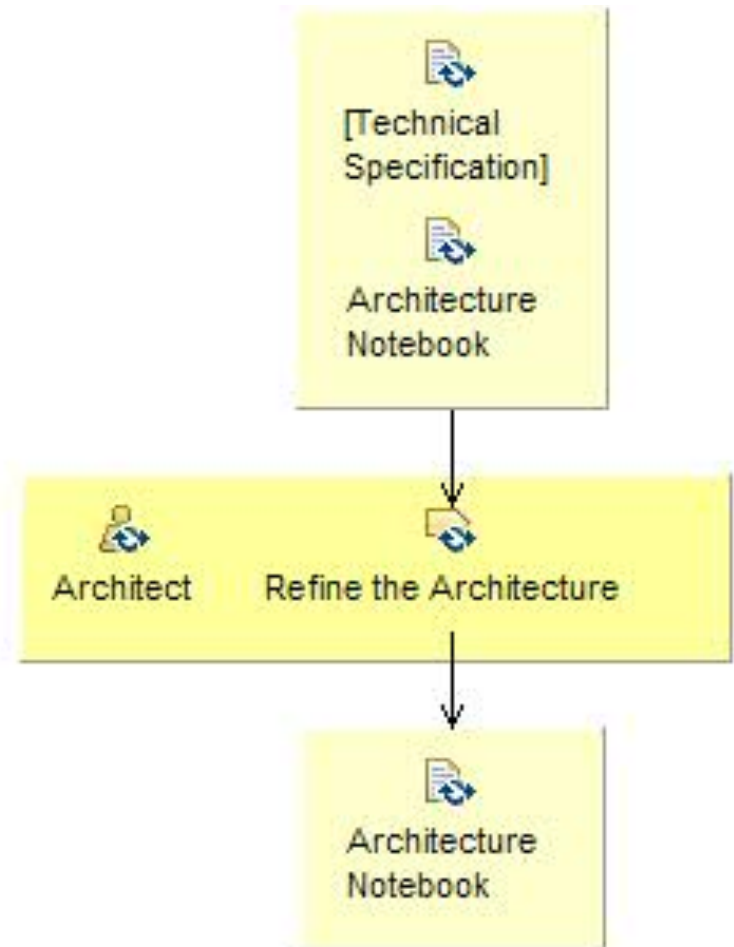
- ▶ Desenvolver os casos de teste e dados de teste para os requisitos a serem testados
 - Atingir um entendimento compartilhado sobre as condições específicas que o sistema deve atender
- ▶ Passos:
 1. Revisar os requisitos a serem testados
 2. Identificar casos de teste relevantes
 3. Esboçar os casos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

Atividades a Serem Executadas



Desenvolver a Arquitetura

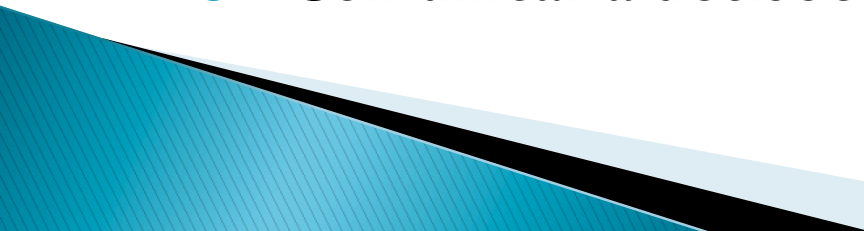
- ▶ Desenvolver os requisitos arquiteturalmente significantes priorizados para a iteração
- ▶ Refina a arquitetura inicial de alto nível em um software funcionando
- ▶ Com o objetivo principal de produzir um software estável que atenda aos riscos técnicos no escopo



Desenvolver a Arquitetura

- ▶ Arquitetos e desenvolvedores trabalham juntos para:
 - Refinar o esboço inicial da arquitetura em elementos de projeto concreto
 - Garantir que as decisões arquiteturais foram adequadamente capturadas e comunicadas
 - Garantir que a equipe possui informações suficientes para permitir que o software seja desenvolvido
 - Garantir que os requisitos que foram priorizados para a iteração corrente foram adequadamente atendidos no software

Tarefa: Refinar a Arquitetura

- ▶ Implementar, avaliar e evoluir com a arquitetura executável
 - ▶ Passos:
 1. Refinar as metas arquiteturais e os requisitos arquiteturalmente significantes
 2. Identificar elementos de projeto arquiteturalmente significantes (componentes, classes e subsistemas)
 3. Refinar os mecanismos arquiteturais
 4. Definir o desenvolvimento e o teste da arquitetura
 5. Identificar oportunidades adicionais de reuso
 6. Validar a arquitetura
 7. Mapear o software ao hardware
 8. Comunicar a decisões
- 



Definindo a Arquitetura

- ▶ Definições baseadas na abordagem *Agile Modeling* (Scott W. Ambler)
 - <http://www.agilemodeling.com>
- ▶ Razões para fazer no início do projeto uma modelagem ágil da arquitetura:
 1. Aumentar a produtividade
 2. Reduzir o risco técnico
 3. Reduzir o tempo de desenvolvimento
 4. Melhorar a comunicação
 5. Subsidiando o desenvolvimento ágil de software
 6. Melhorar a organização da equipe

Definindo a Arquitetura

- ▶ O que deve ser modelado inicialmente?
 1. Diagramas de tecnologia
 2. Fluxo da interface com o usuário
 3. Modelos de domínio
 4. Casos de mudança

Diagrama de Tecnologia

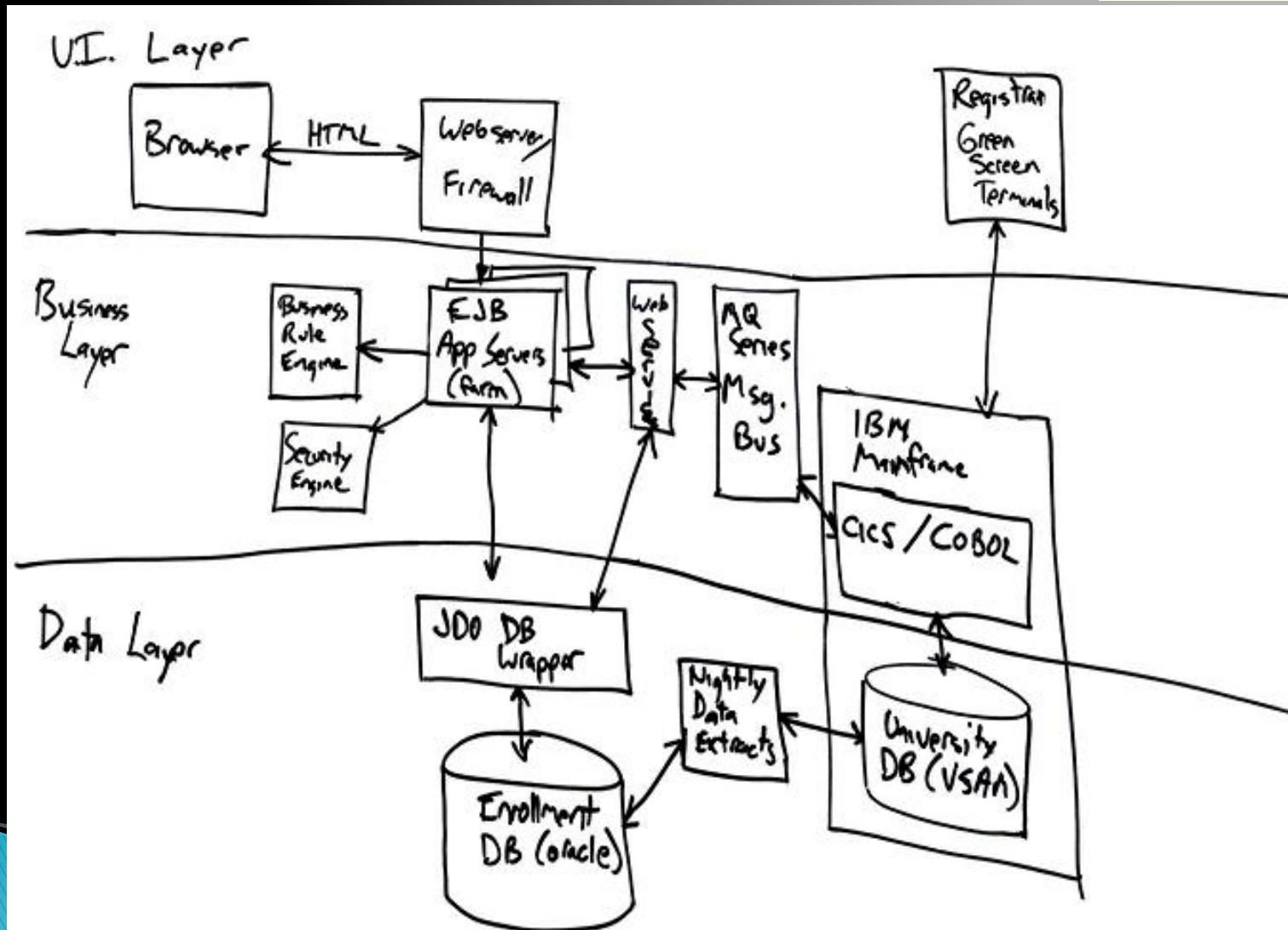
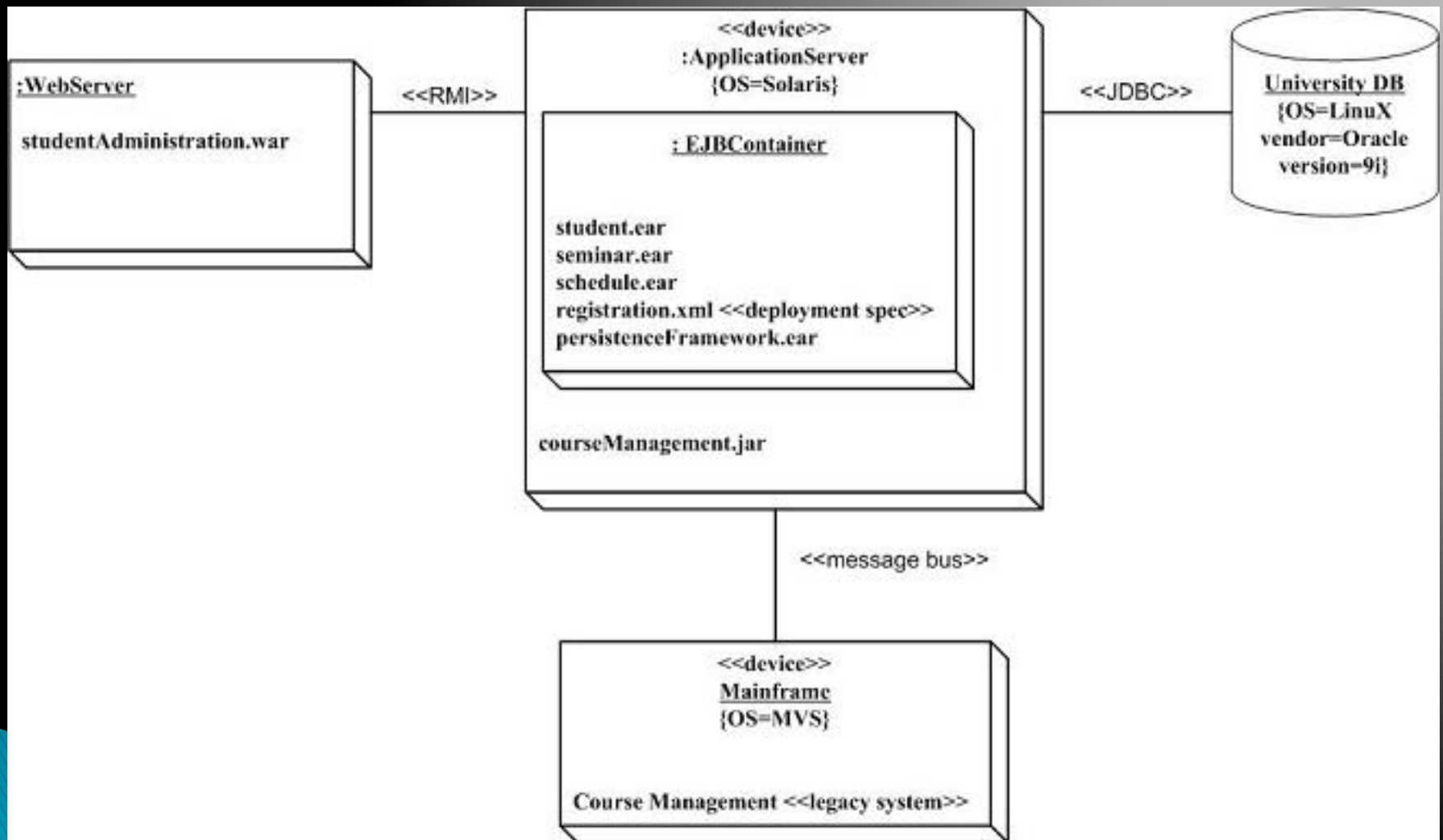
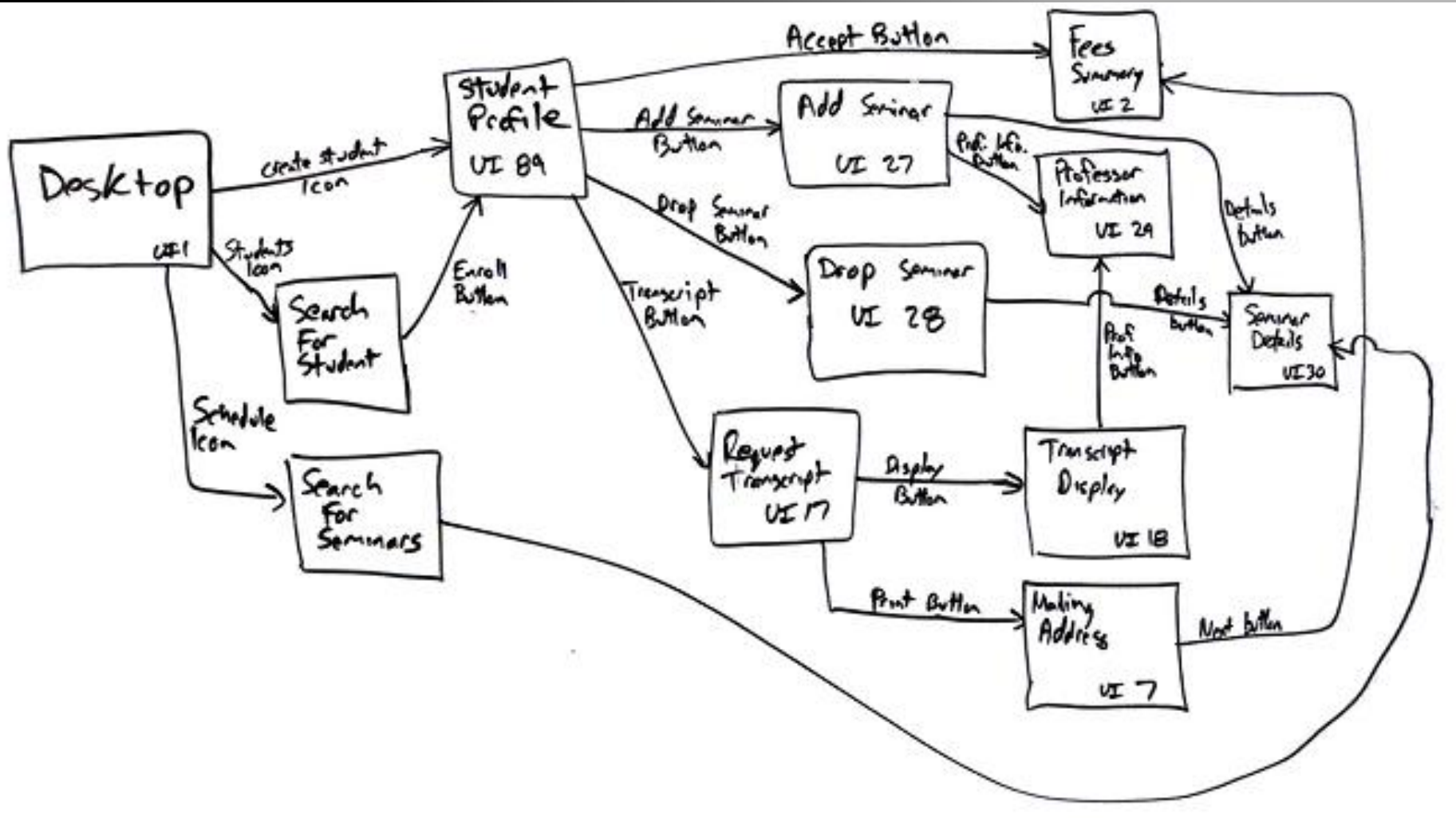


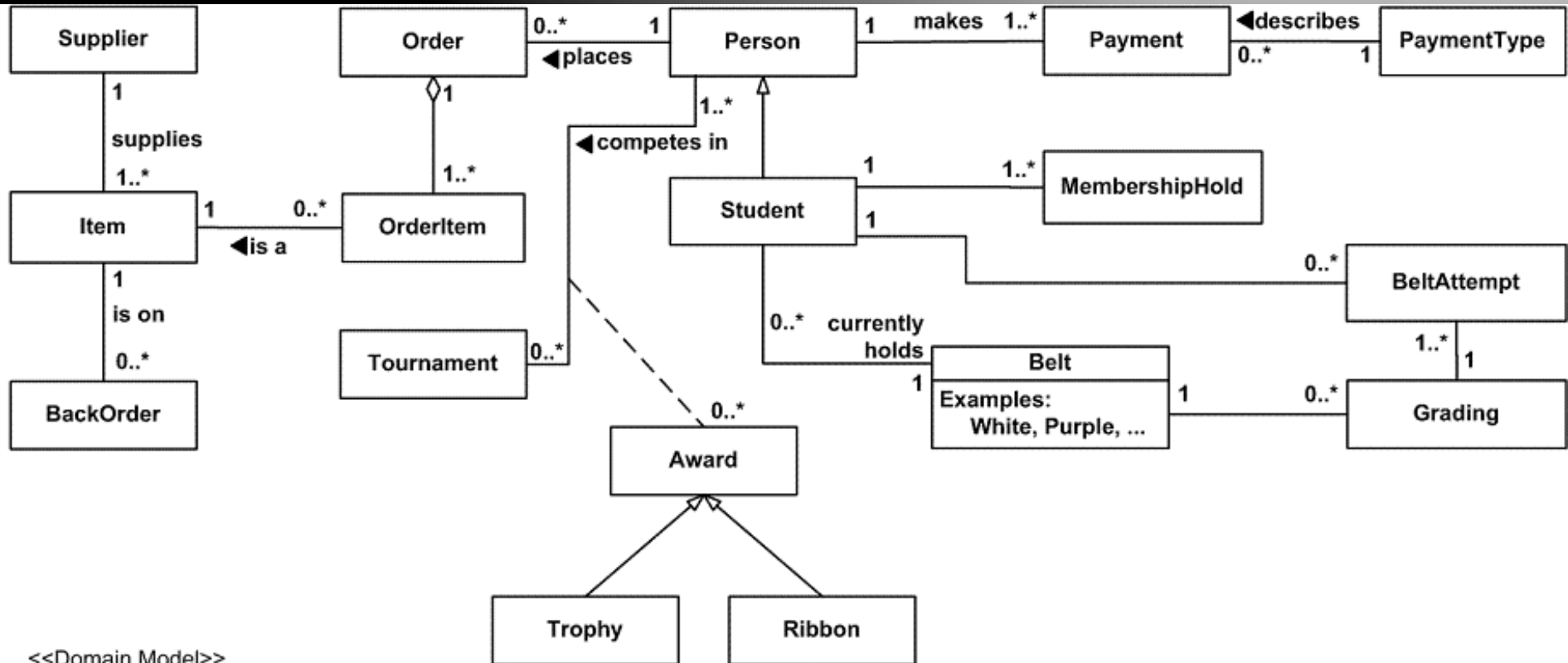
Diagrama de Tecnologia



Modelo de Fluxo de UI



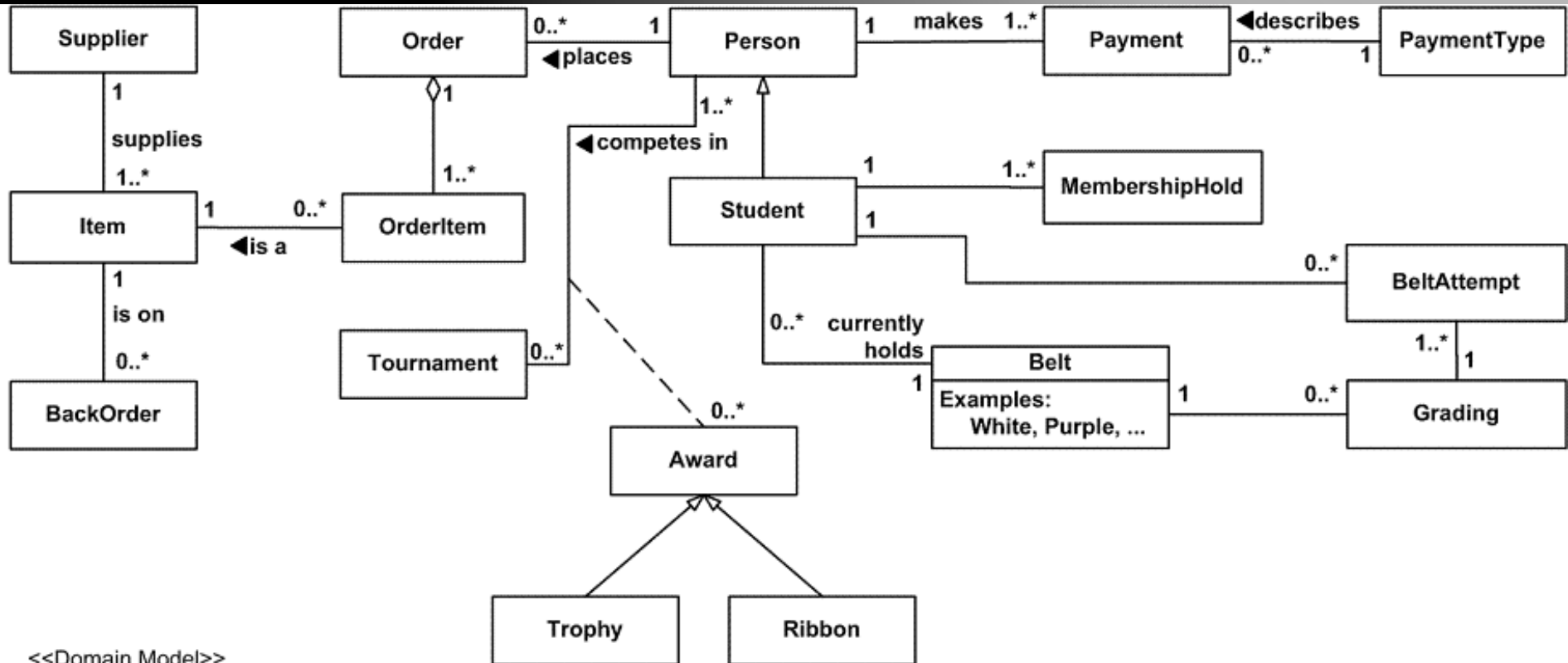
Modelo de Domínio



<<Domain Model>>

Copyright 2004 Scott W. Ambler

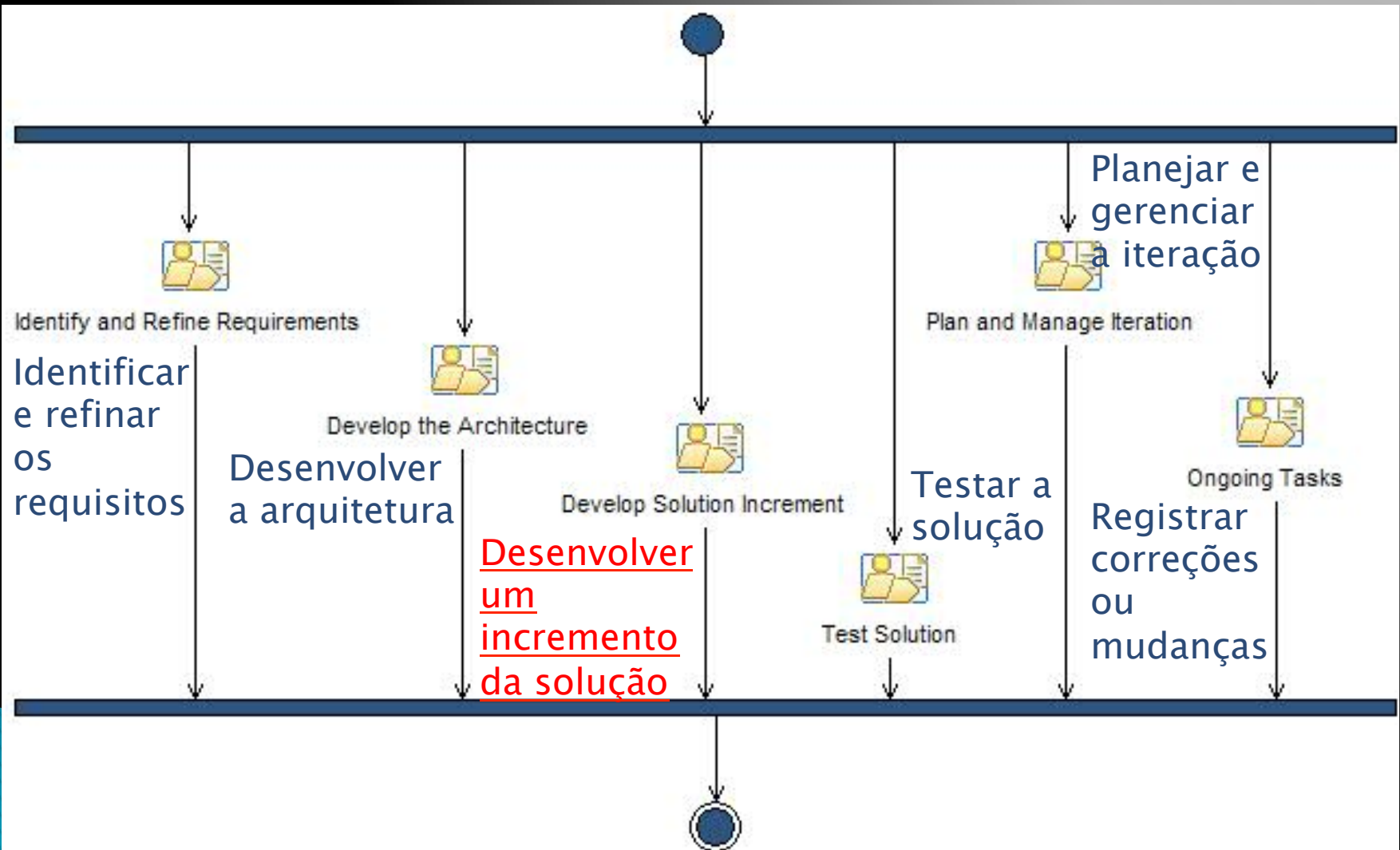
Modelo de Domínio



<<Domain Model>>

Copyright 2004 Scott W. Ambler

Atividades a Serem Executadas

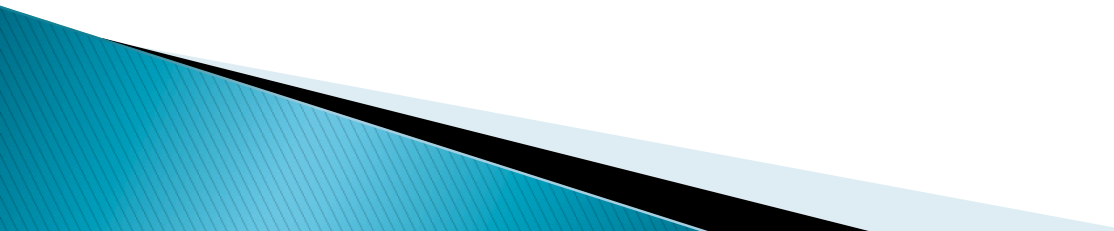


Desenvolver um Incremento da Solução

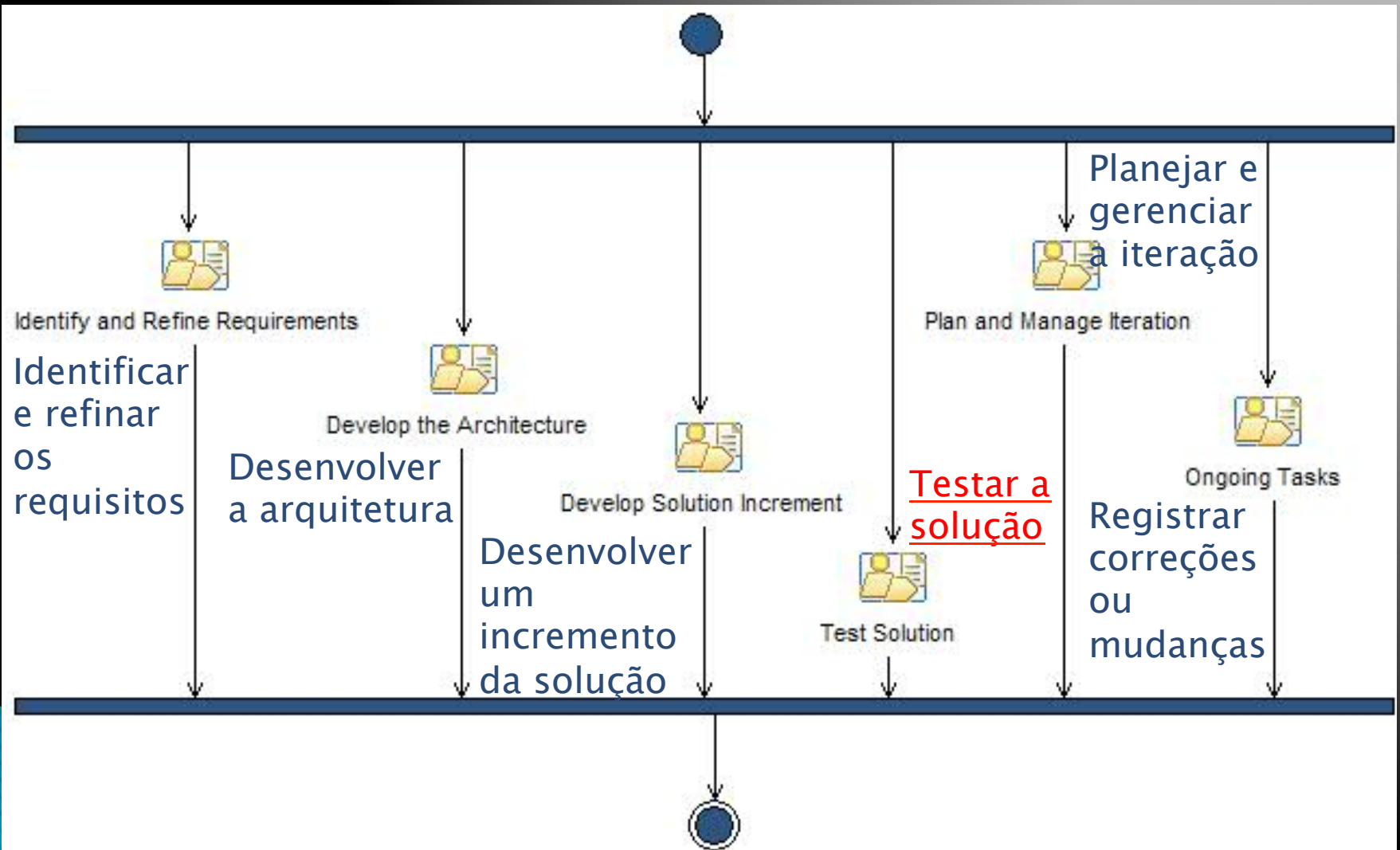
- ▶ Forma de executar o planejamento e execução baseados em metas
- ▶ Facilita o acompanhamento e o *feedback*



Ordem das Tarefas

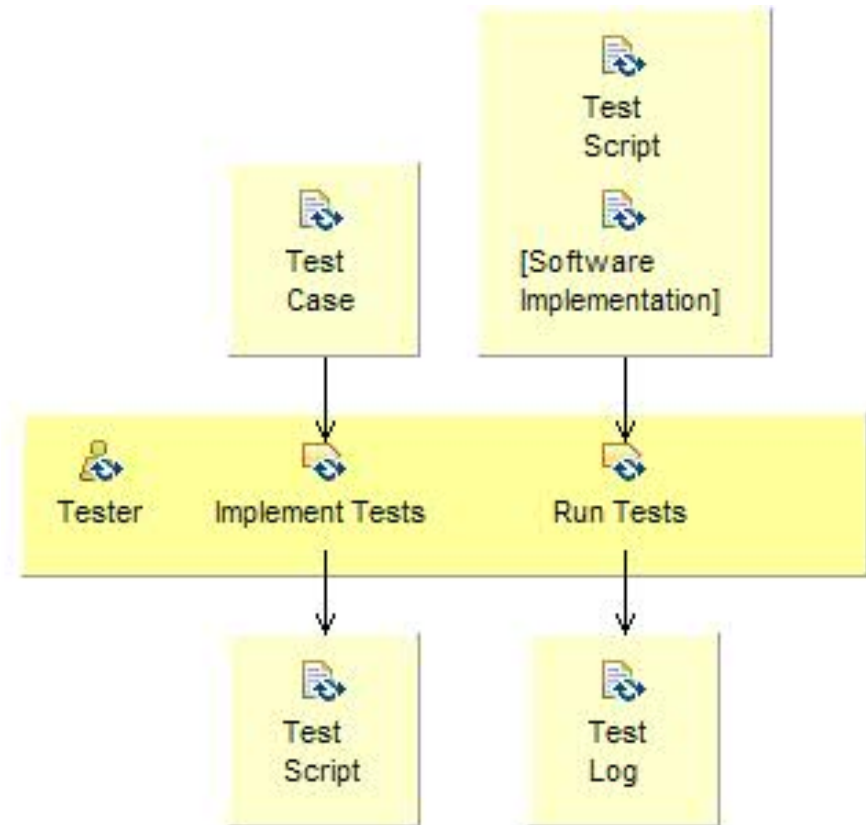
1. Projetar a solução
 - Diagramas de interação, [de estados], [de atividades], e de classes de projeto
 2. Implementar os testes de desenvolvedor
 - Implementar os teste unitários (teste de classe)
 3. Implementar a solução
 - Codificar segundo os padrões de codificação
 4. Executar os testes de desenvolvedor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

Atividades a Serem Executadas



Testar a Solução

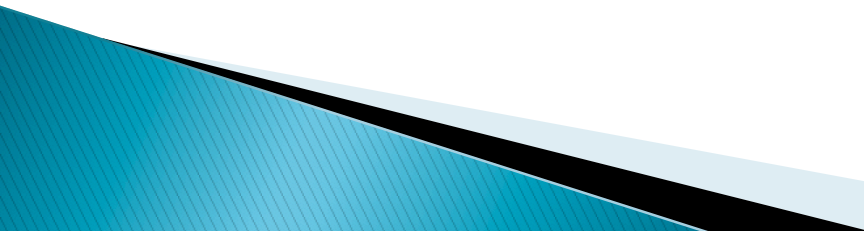
- ▶ Desenvolver e executar testes que garantem que o sistema satisfaz aos requisitos
- ▶ Validar se os requisitos implementados refletem uma arquitetura robusta



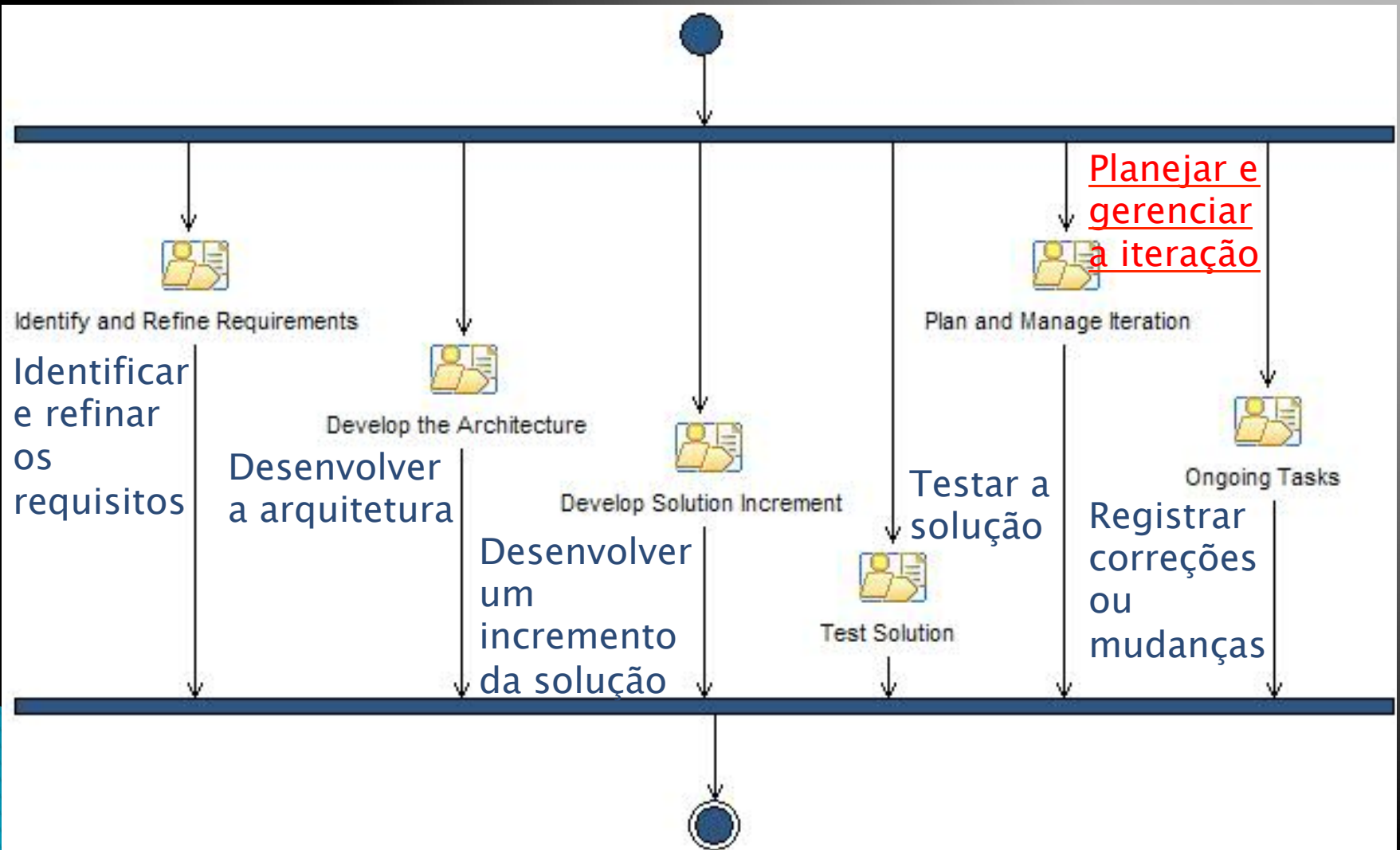
Tarefa: Implementar os Teste

- ▶ 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

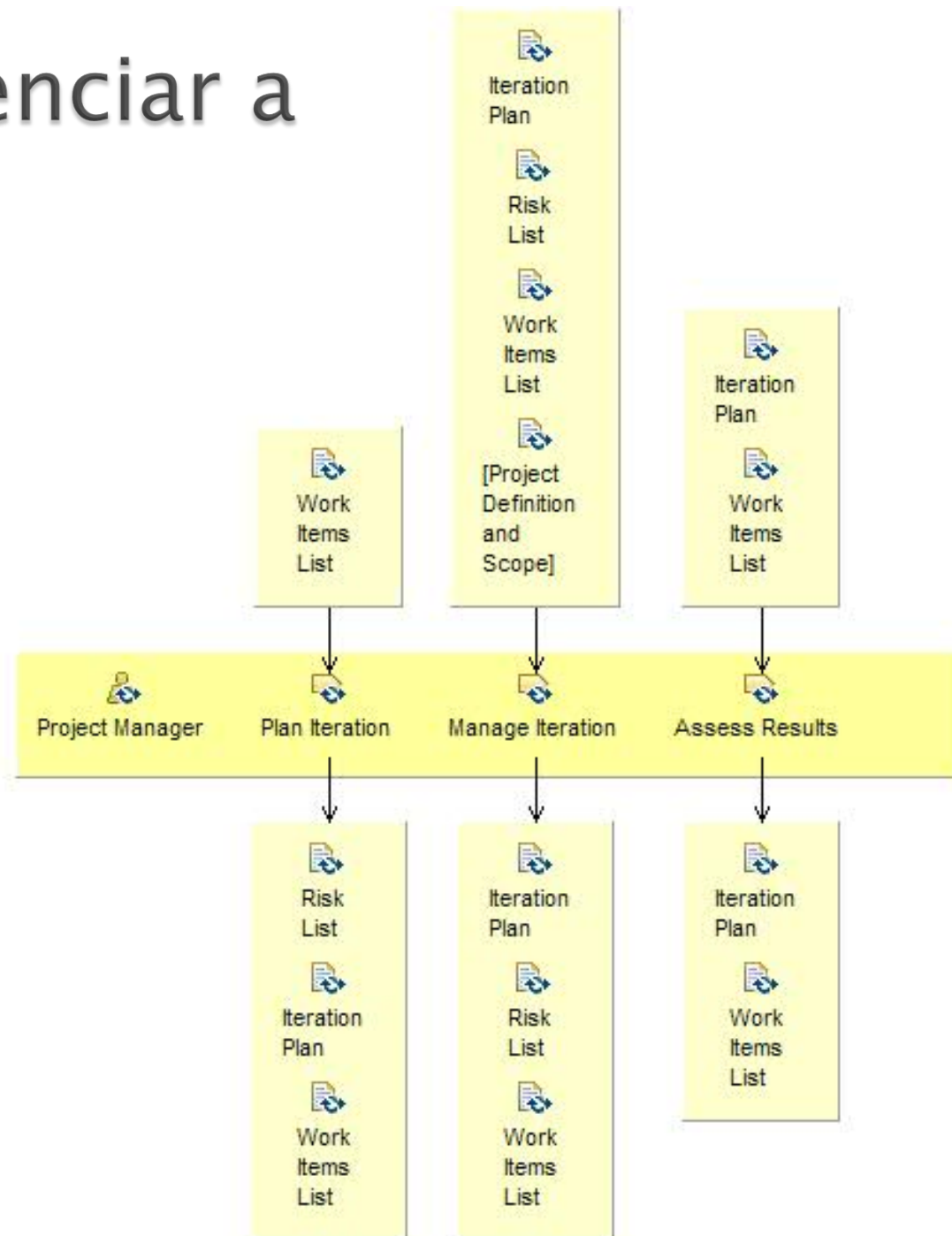
- ▶ Prover um feedback para a equipe de quanto bem o *build* satisfaz os requisitos selecionados
 - ▶ Passos:
 1. Rever os itens de trabalho completados para o *build*
 2. Selecionar os roteiros de testes
 3. Executar os roteiros de testes no *build*
 4. Analisar e comunicar os resultados dos testes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

Atividades a Serem Executadas

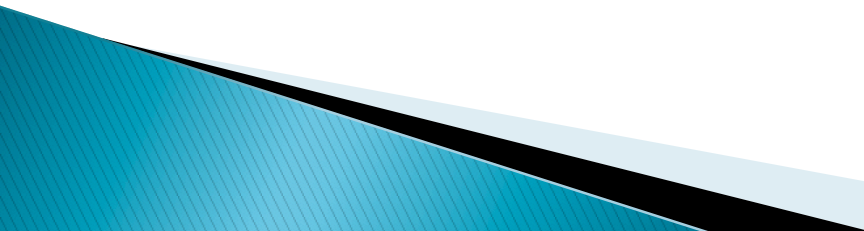


Planejar e Gerenciar a Iteração

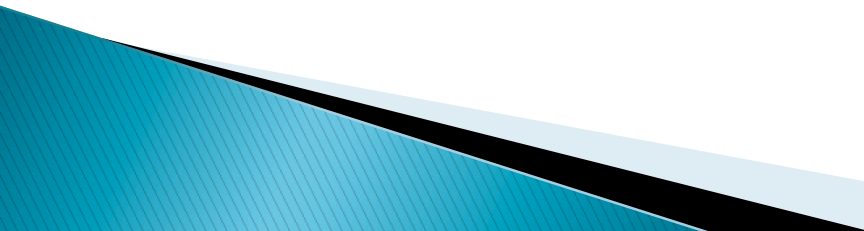
- ▶ Iniciar a iteração e atribuir tarefas de desenvolvimento aos membros da equipe
- ▶ Monitorar e informar o status para os *stakeholders*
- ▶ Identificar e gerenciar as exceções e problemas



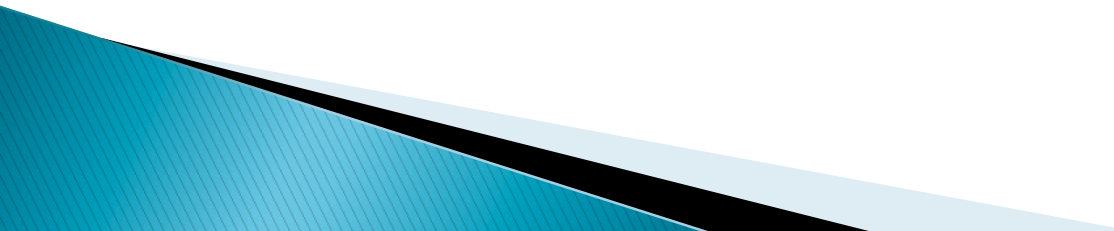
Tarefa: Planejar a Iteração

- ▶ 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
- 

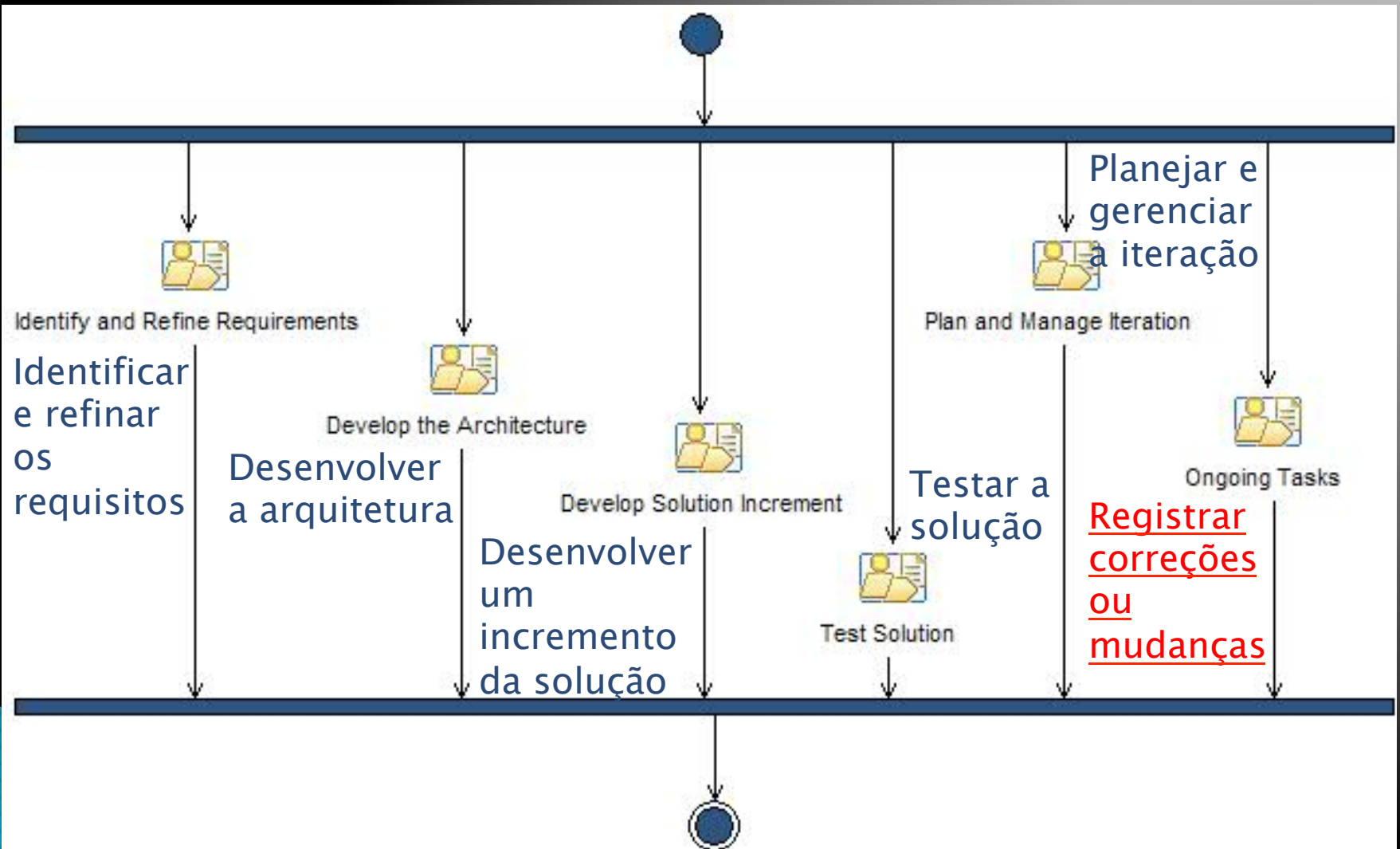
Tarefa: Gerenciar a Iteração

- ▶ 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

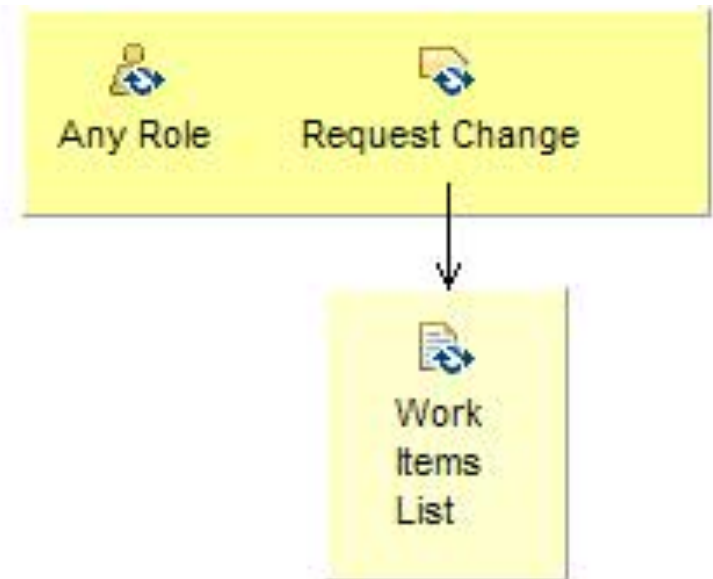
- ▶ Determinar o sucesso ou falha da iteração
 - ▶ Aplicar as lições aprendidas para modificar o projeto e aperfeiçoar o processo
 - ▶ Passos:
 1. Preparar para a avaliação da iteração
 2. Demonstrar o valor e obter feedback
 3. Realizar uma retrospectiva
 4. Realizar uma retrospectiva (final da fase)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

Atividades a Serem Executadas



Registrar Correções ou Mudanças

- ▶ Atividade que não faz parte do cronograma
- ▶ Inclui 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