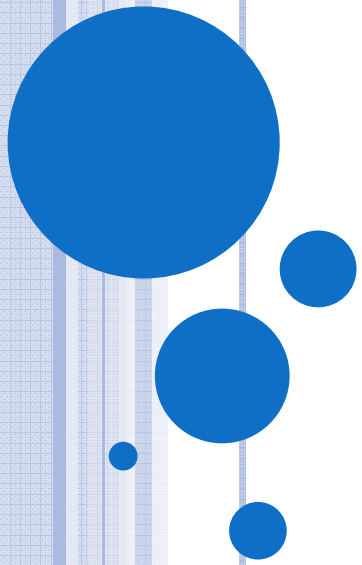


PROCESSOS PRESCRITIVOS

INE 5419 – Engenharia de Software II

Prof. Raul Sidnei Wazlawick

UFSC-CTC-INE

2012.1

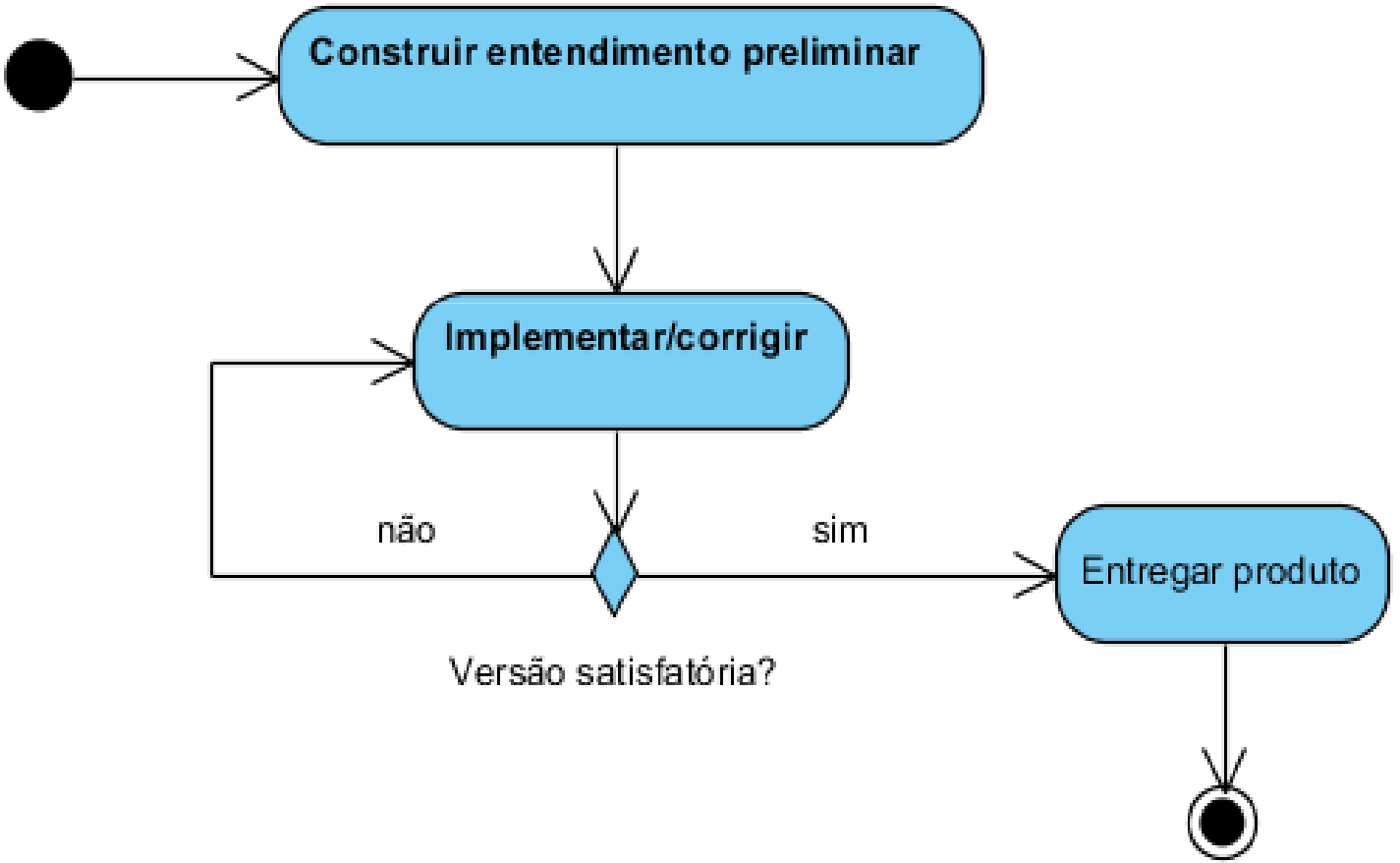
Conteúdo

- Codificar e Testar
- Modelo Cascata
- Modelo Sashimi
- Modelo V
- Modelo W
- Modelo Cascata com Subprojetos
- Modelo Cascata com Redução de Risco
- Modelo Espiral
- Prototipação Evolucionária
- Entrega em Estágios
- Modelo Orientado a Cronograma
- Entrega Evolucionária
- Modelos Orientados a Ferramentas
- Linhas de Produto de Software

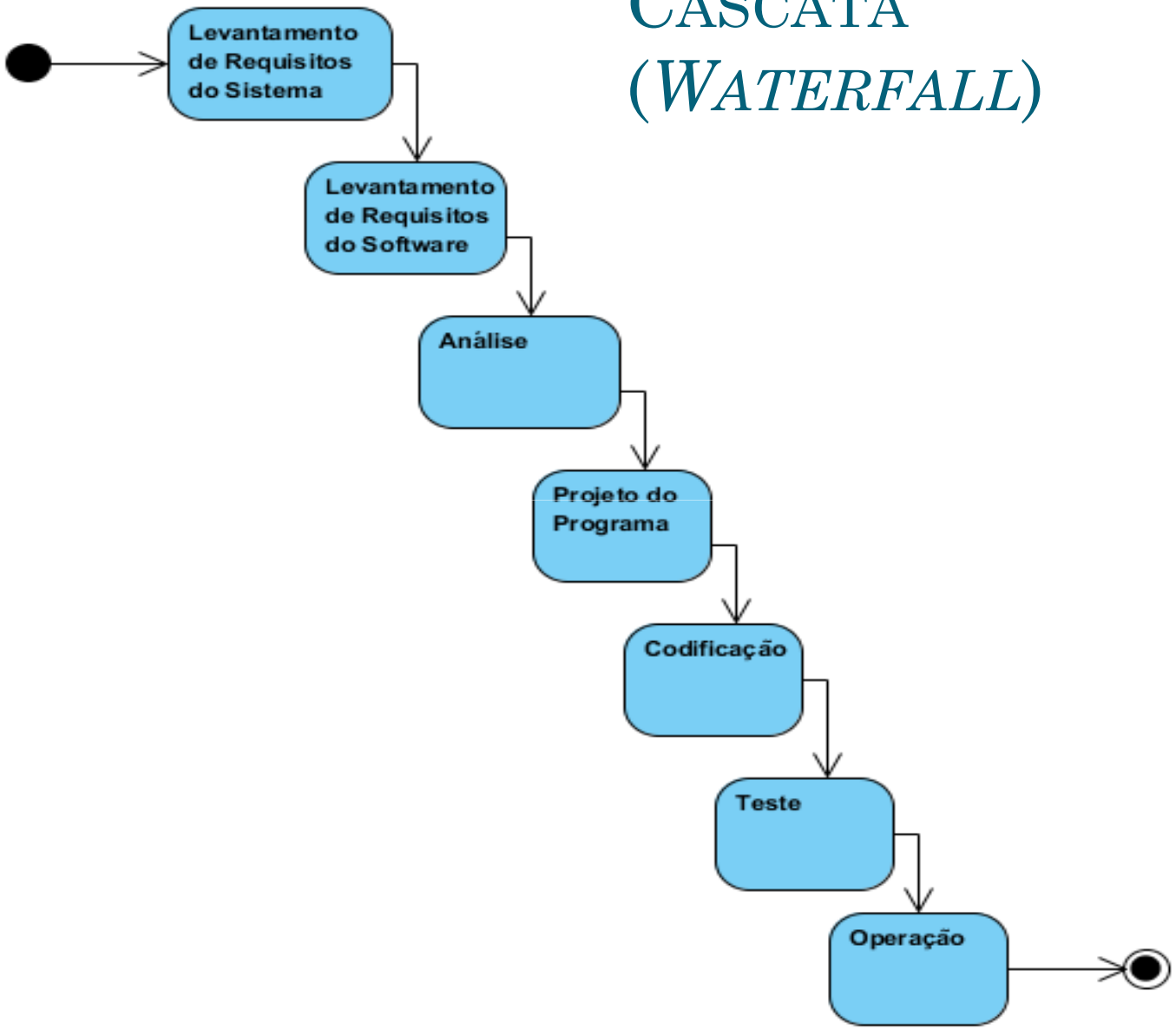
QUESTÕES QUE DETERMINAM A ESCOLHA DE UM MODELO

- Quão bem os analistas e o cliente podem conhecer os requisitos do sistema?
- Quão bem é compreendida a arquitetura do sistema?
- Qual o grau de confiabilidade necessário em relação ao cronograma?
- Quanto planejamento é efetivamente necessário?
- Qual o grau de risco que este projeto apresenta?
- Existe alguma restrição de cronograma?
- Será necessário entregar partes do sistema funcionando antes de terminar o projeto todo?
- Qual o grau de treinamento e adaptação necessários para a equipe poder utilizar o ciclo de vida que parece mais adequado ao projeto?
- Será desenvolvido um único sistema ou uma família de sistemas semelhantes?
- Qual o tamanho do projeto?

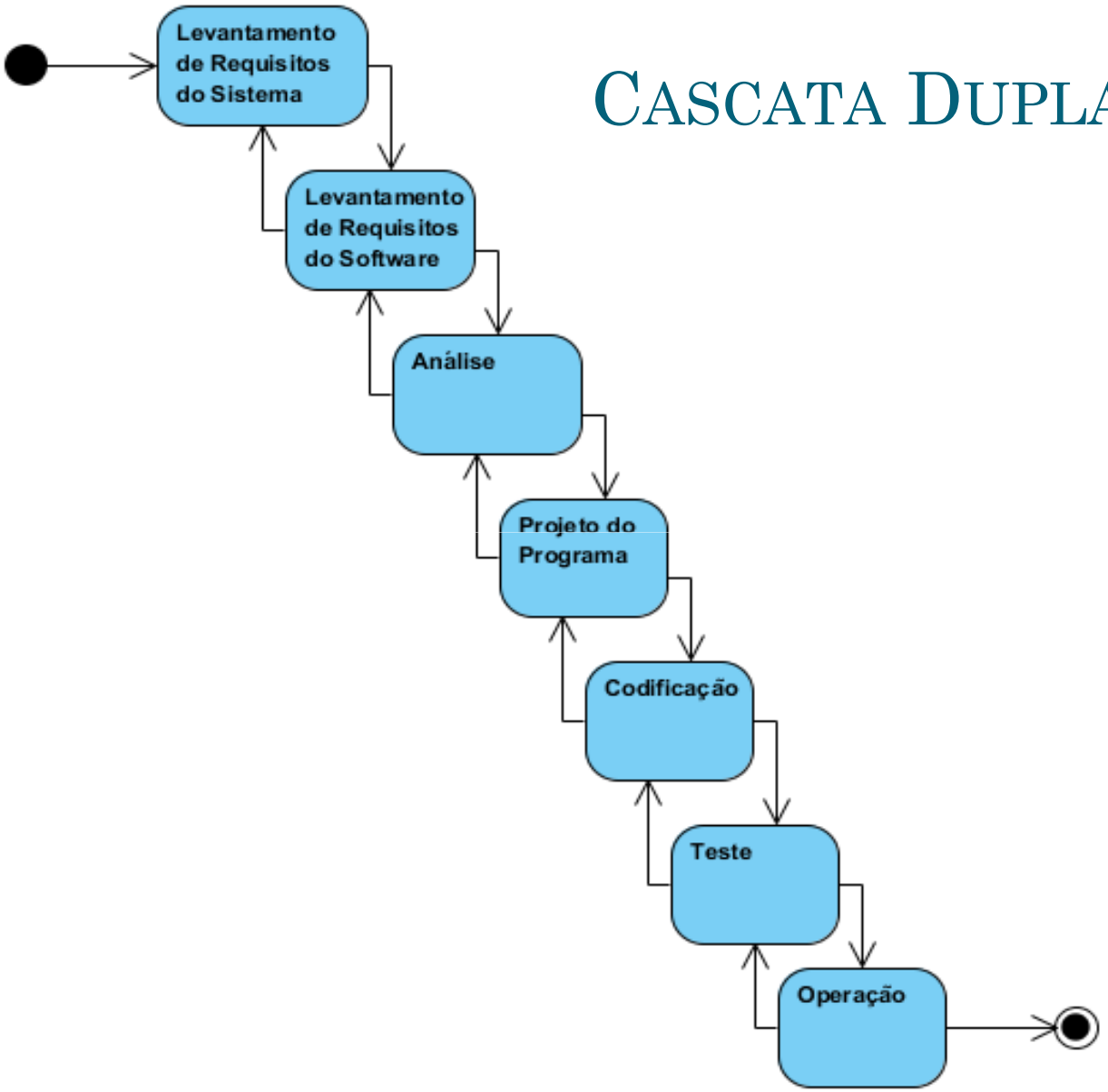
CONDIFICAR E CONSERTAR (*CODE AND FIX*)



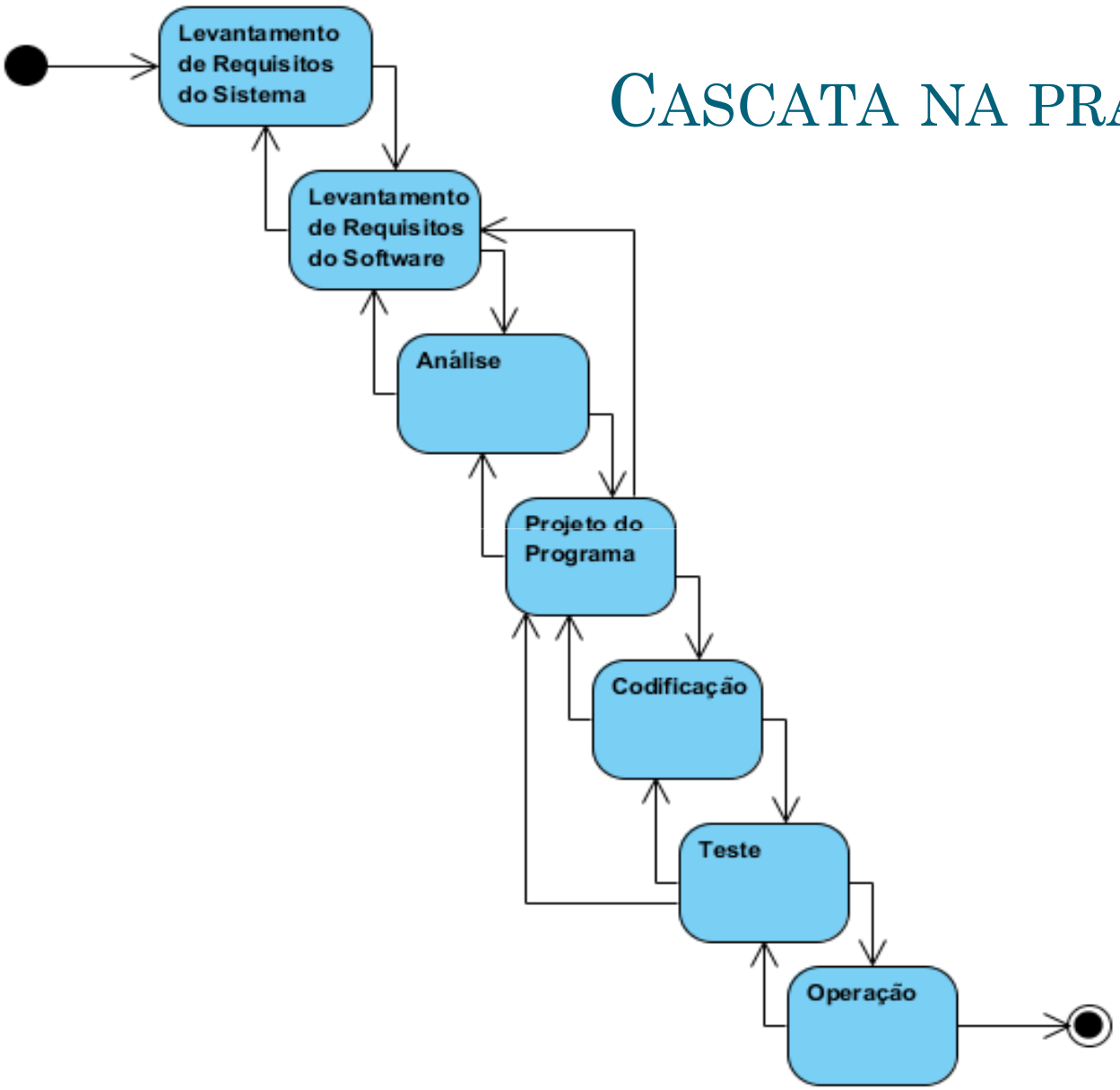
CASCATA (*WATERFALL*)



Cascata Dupla

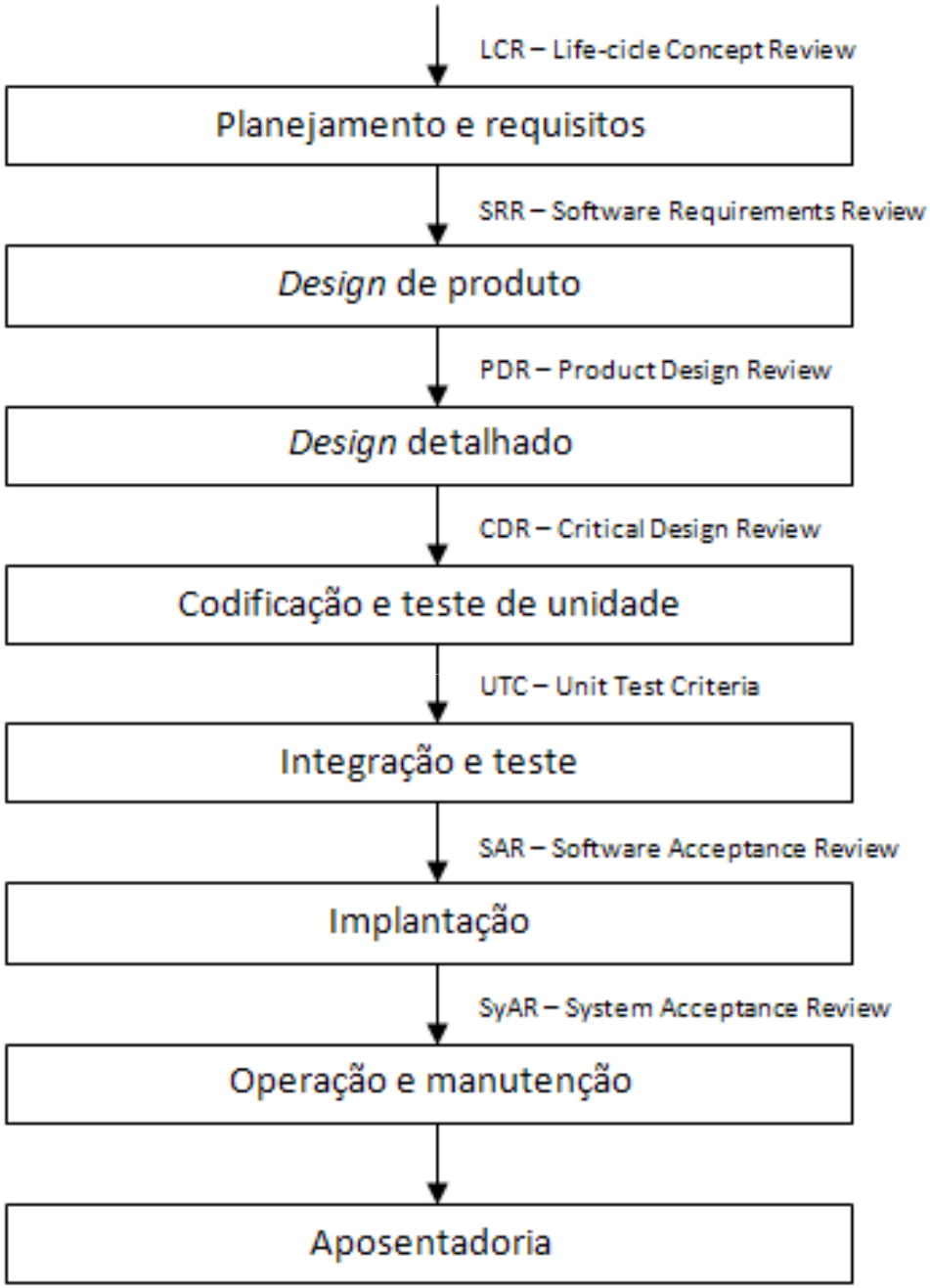


CASCATA NA PRÁTICA



Sugestões de Royce (1970)

- Inserir uma fase de design entre o levantamento dos requisitos e sua análise.
- Produzir documentação.
- Fazer duas vezes.
- Planejar, controlar e monitorar o teste.
- Envolver o cliente.



MARCOS (*MILESTONES*) (BOEHM, 1981)

LCR, Life-cicle Concept Review

- Arquitetura de sistema aprovada e validada, incluindo questões básicas de hardware e software.
- Conceito de operação aprovado e validado, incluindo questões básicas de interação humano-computador.
- Plano de ciclo de vida de alto nível, incluindo marcos, recursos, responsabilidades, cronogramas e principais atividades.

SRR, Software Requirements Review

- Plano de desenvolvimento detalhado:
  - detalhamento de critérios de desenvolvimento de marcos, orçamento e alocação de recursos, organização da equipe, responsabilidades, cronograma, atividades, técnicas e produtos a serem usados.
- Plano de uso detalhado:
  - contraparte para os itens do plano de desenvolvimento como treinamento, conversão, instalação, operações e suporte.
- Plano de controle de produto detalhado:
  - plano de gerenciamento de configuração, plano de garantia de qualidade, plano geral de V&V (verificação e validação), excluindo detalhes dos planos de testes.
- Especificações de requisitos de software aprovadas e validadas:
  - requisitos funcionais, de performance e especificações de interfaces validadas em relação a completude, consistência, testabilidade e exequibilidade.
- Contrato de desenvolvimento (formal ou informal) aprovado baseado nos itens acima.

PDR, Product Design Review

- Especificação do design do produto de software verificada.
- Hierarquia de componentes do programa, interfaces de controle e dados entre as unidades (uma *unidade* de software realiza uma função bem definida, pode ser desenvolvida por uma pessoa e tipicamente tem de 100 a 300 linhas de código).
- Estruturas de dados lógicas e físicas detalhadas em nível de seus campos.
- Orçamento para recursos de processamento de dados (incluindo especificações de eficiência de tempo, capacidade de armazenamento e precisão).
- Verificação do *design* com referência a completude, consistência, exequibilidade e rastreabilidade aos requisitos.
- Identificação e resolução de todos os riscos de alta importância.
- Plano de teste e integração preliminar, plano de teste de aceitação e manual do usuário.

CDR, Critical Design Review

- Especificação de *design* detalhado revisada para cada unidade.
- Para cada rotina (menos de 100 instruções) dentro de uma unidade, espeficicar nome, propósito, hipóteses, tamanho, sequencia de chamadas, entradas, saídas, exceções, algoritmos e fluxo de processamento.
- Descrição detalhada da base de dados.
- Especificações e orçamentos de *design* verificados em relação a completude, consistência e rastreabilidade em relação aos requisitos.
- Plano de teste de aceitação aprovado.
- Manual do usuário, e rascunho do plano de teste e integração completados.

UTC, Unit Test Criteria

- Verificação de todas as unidades de computação usando não apenas valores nominais, mas também valores singulares e extremos.
- Verificação de todas as entradas e saídas unitárias, incluindo mensagens de erro.
- Exercitar todos os procedimentos executáveis e todas as condições de teste.
- Verificação de conformação a padrões de programação.
- Documentação em nível de unidade completada.

SAR, Software Acceptance Review

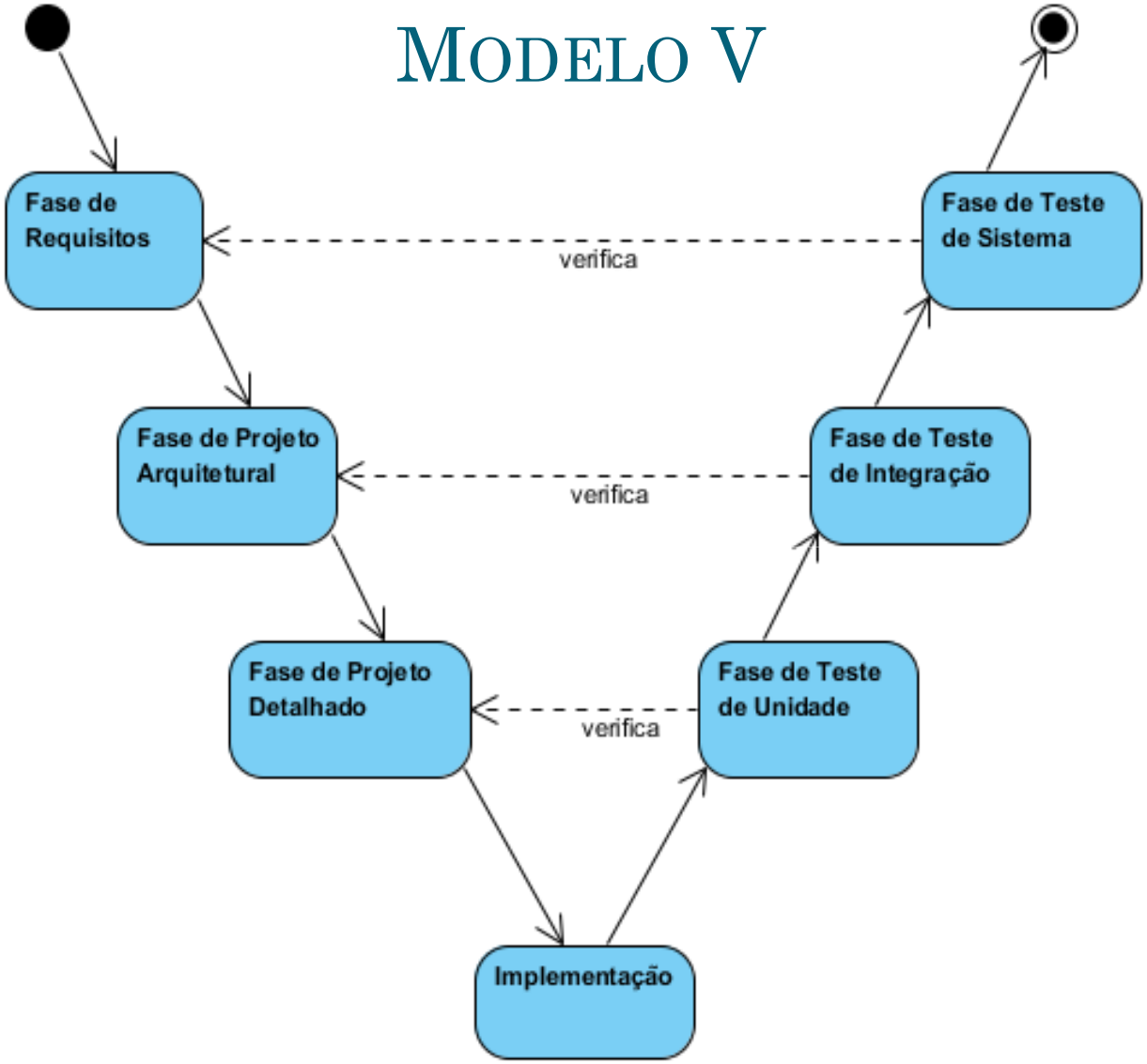
- Testes de aceitação do software satisfeitos.
- Verificação da satisfação dos requisitos do software.
- Demonstração de performance aceitável acima do nominal, conforme especificado.
- Aceitação de todos os produtos do software: relatórios, manuais, especificações e bases de dados.

SyAR, System Acceptance Review

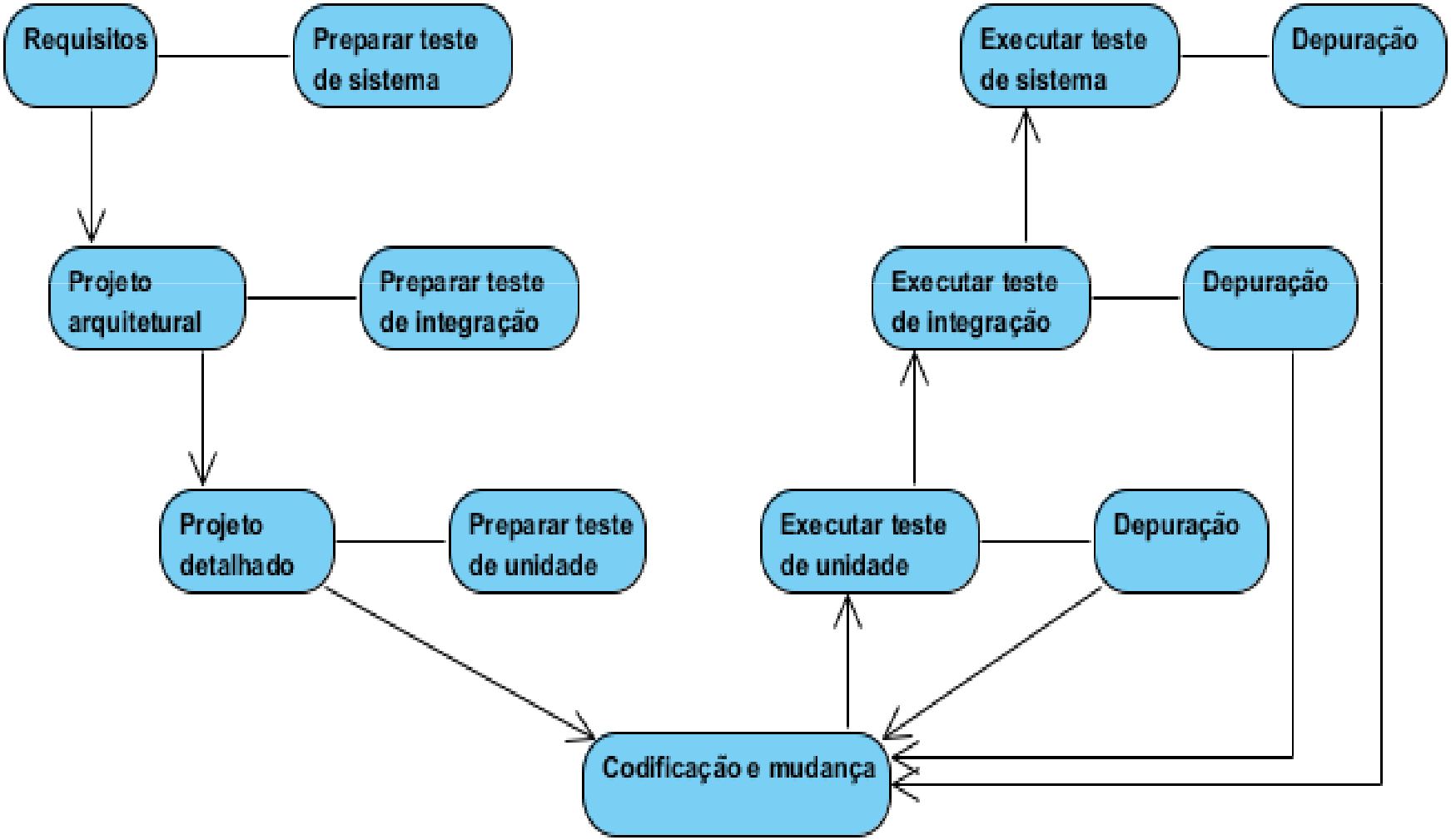
- Satisfação do teste de aceitação do sistema.
- Verificação da satisfação dos requisitos do sistema.
- Verificação da prontidão operacional do software, hardware, instalações e pessoal.
- Aceitação de todas as entregas relacionadas ao sistema: hardware, software, documentação, treinamento e instalações.
- Todas as conversões especificadas e atividades de instalação foram completadas.

MODELO SASHIMI

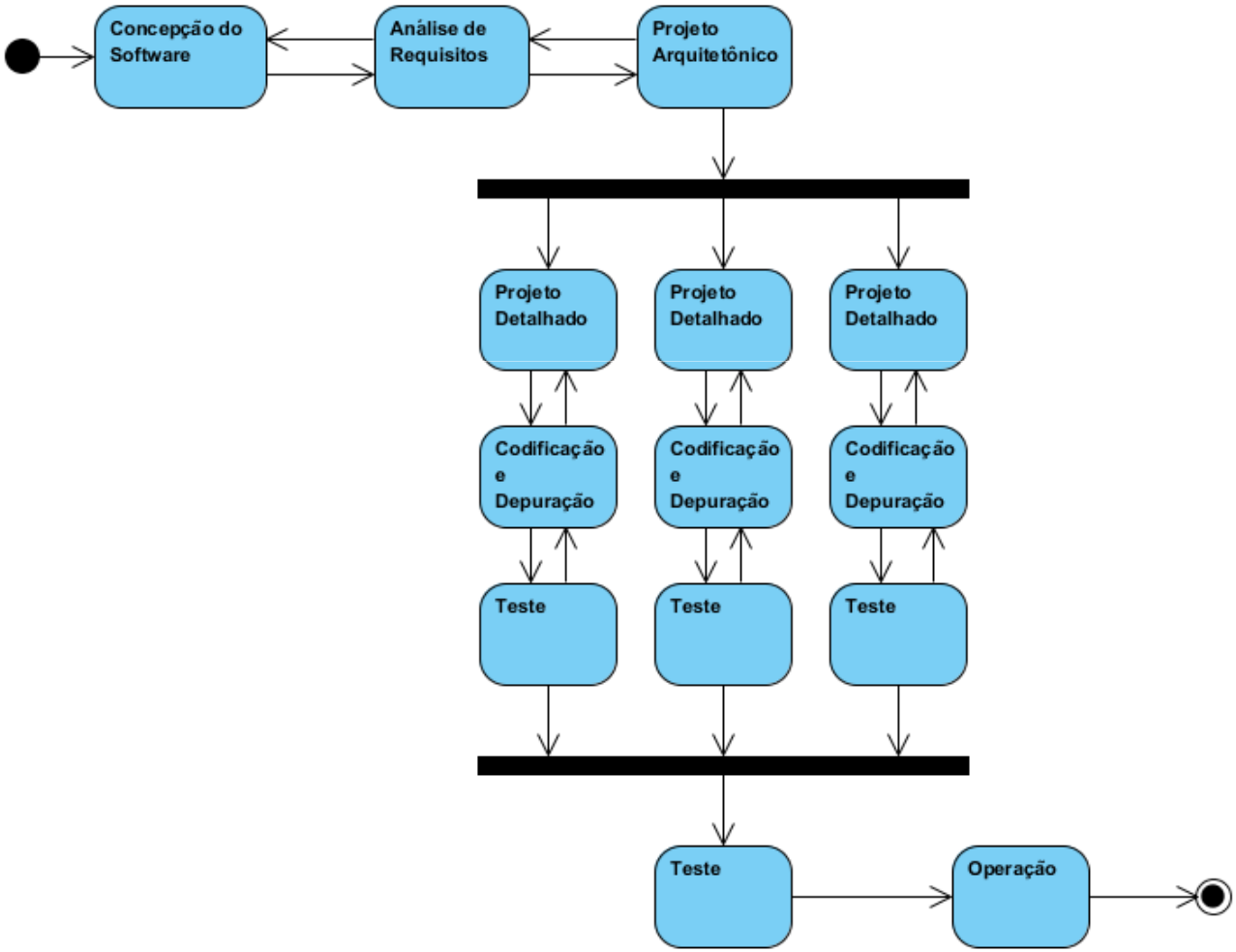
Modelo V


Fase de Requisitos
verifica
Fase de Teste de Sistema
Fase de Projeto Arquitetural
verifica
Fase de Teste de Integração
Fase de Projeto Detalhado
verifica
Fase de Teste de Unidade
Implementação

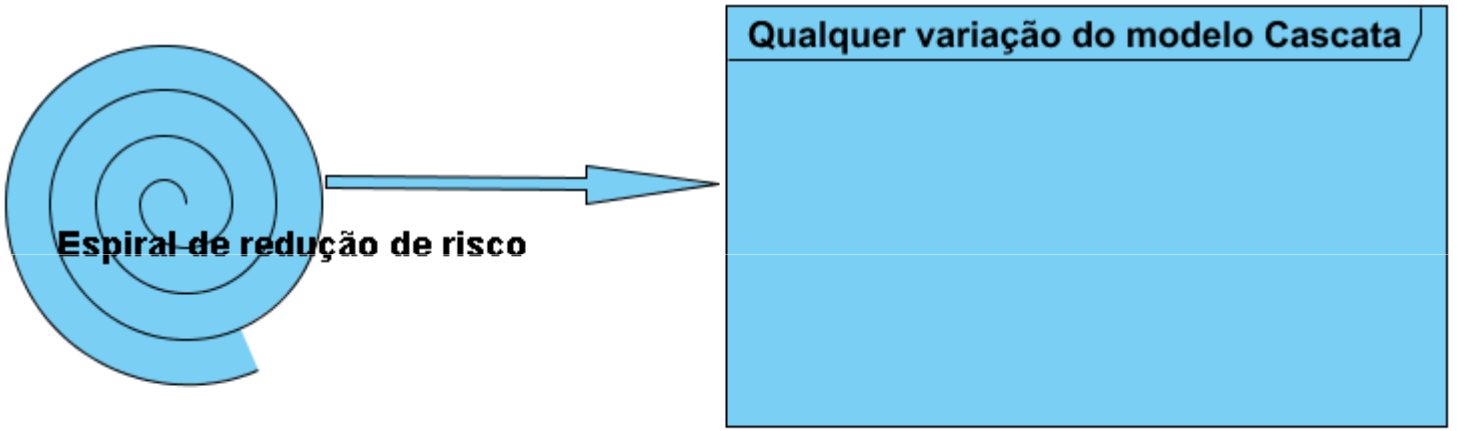
MODELO W


Requisitos
Preparar teste de sistema
Executar teste de sistema
Depuração
Projeto arquitetural
Preparar teste de integração
Executar teste de integração
Depuração
Projeto detalhado
Preparar teste de unidade
Executar teste de unidade
Depuração
Codificação e mudança

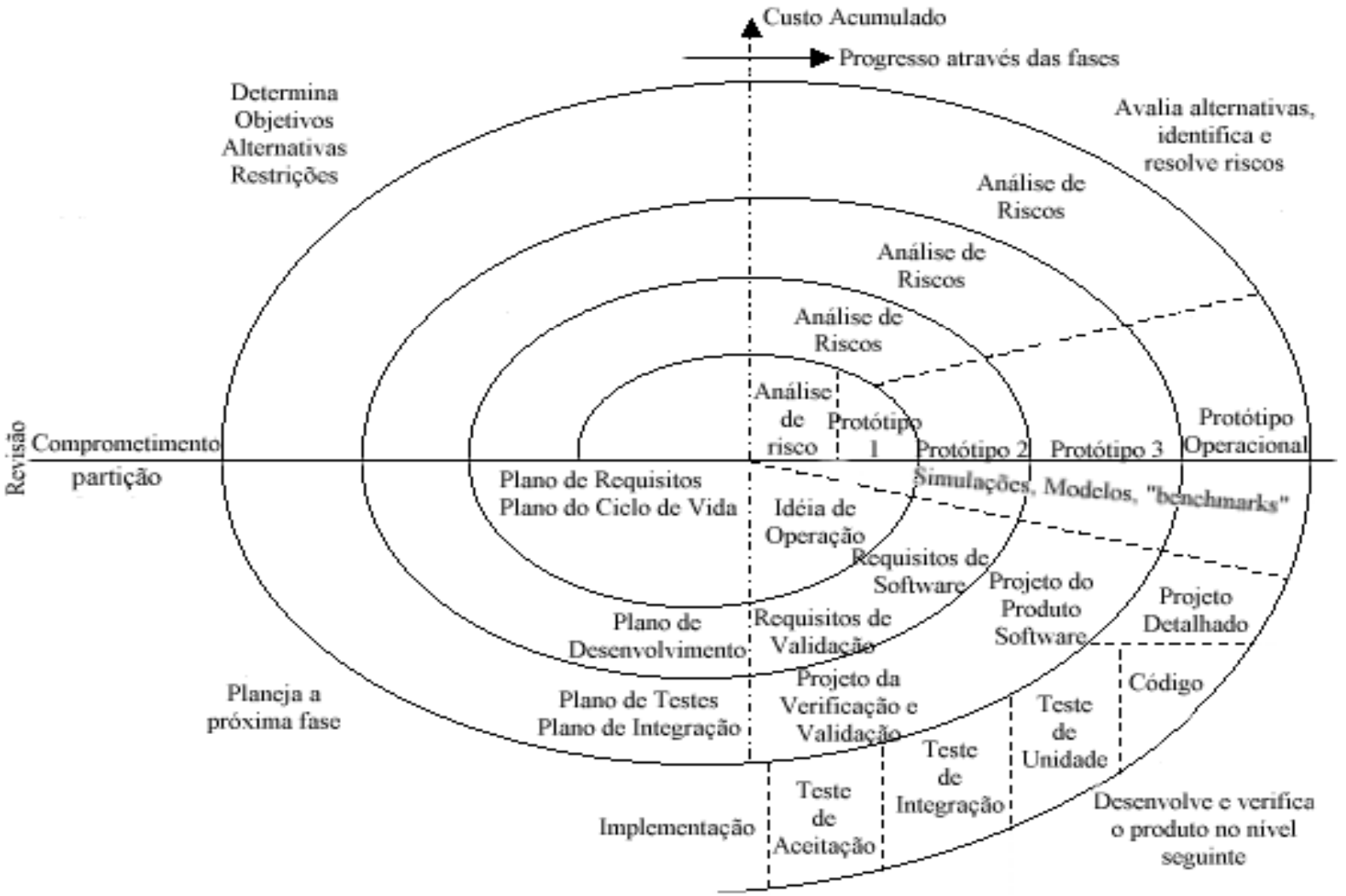
CASCATA COM SUBPROJETOS



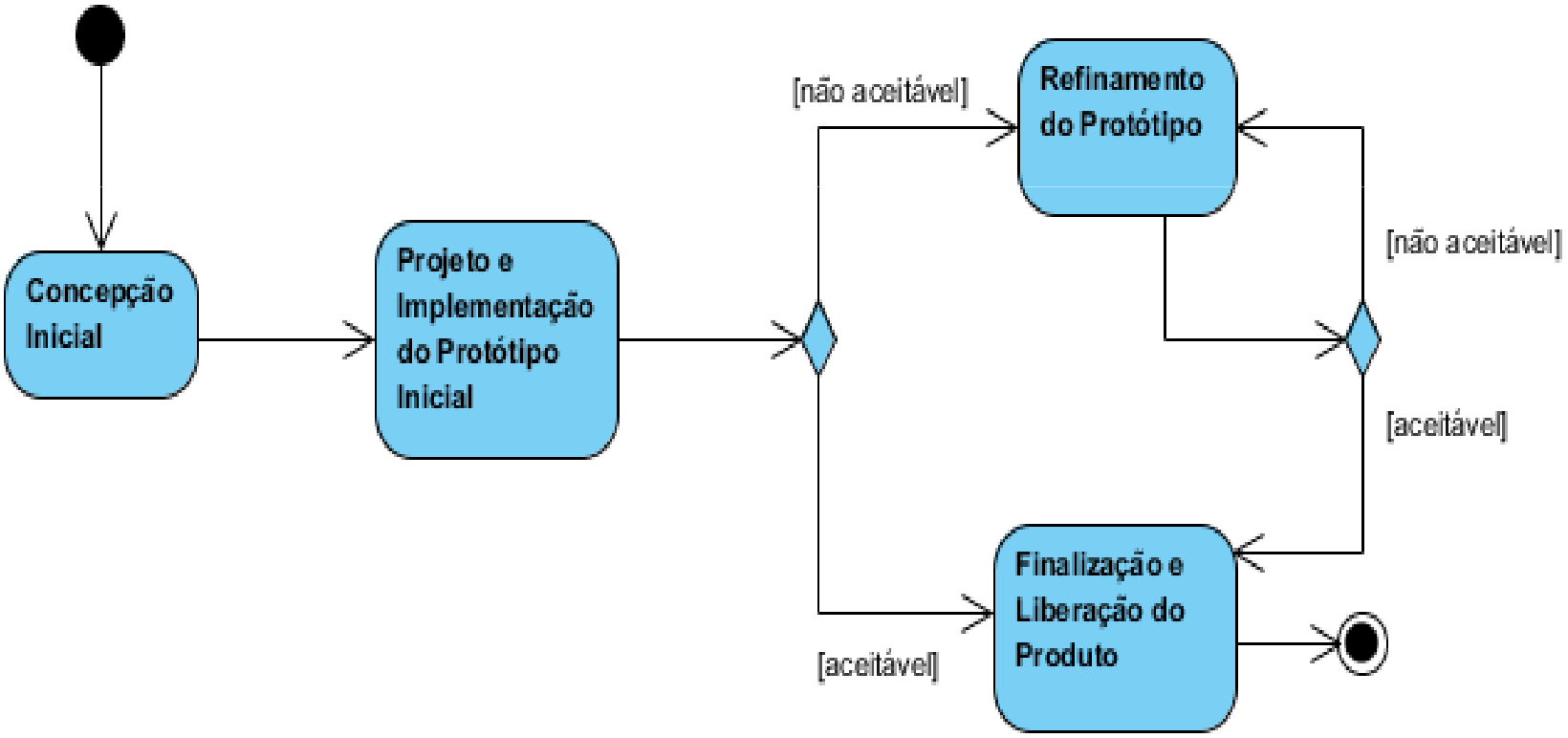
CASCATA COM REDUÇÃO DE RISCO



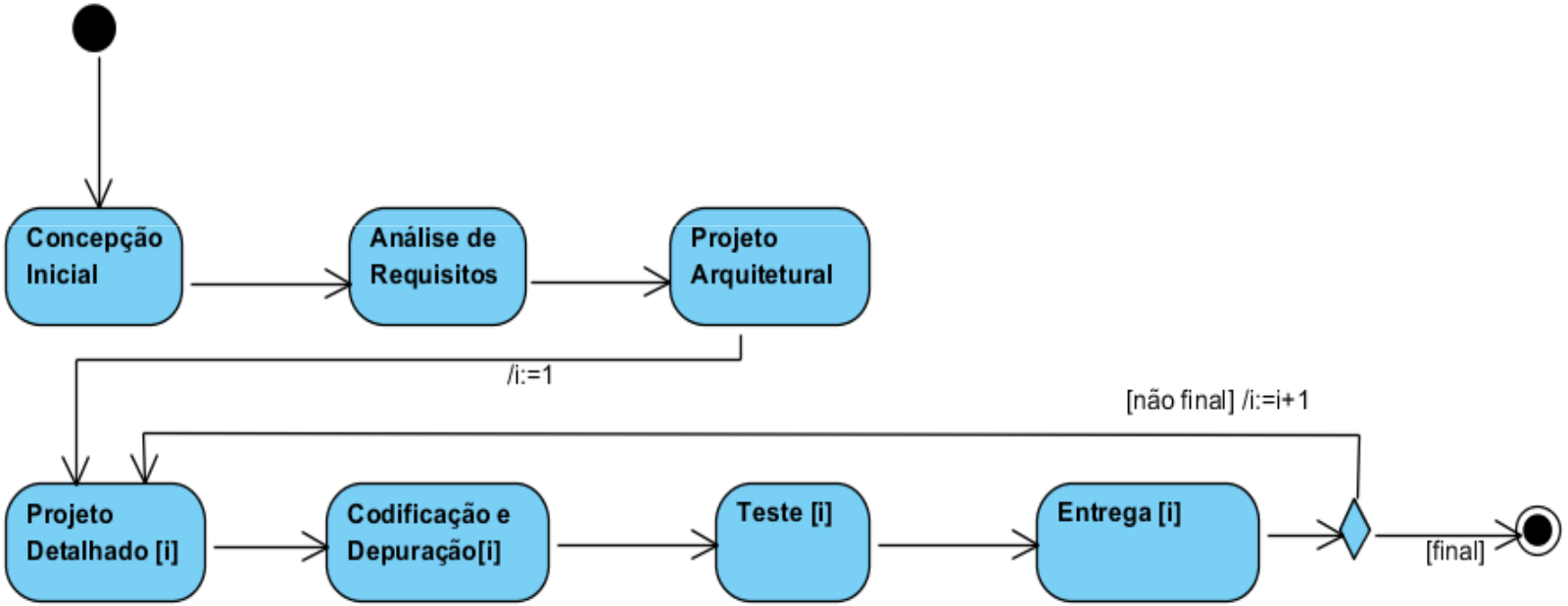
Modelo Espiral



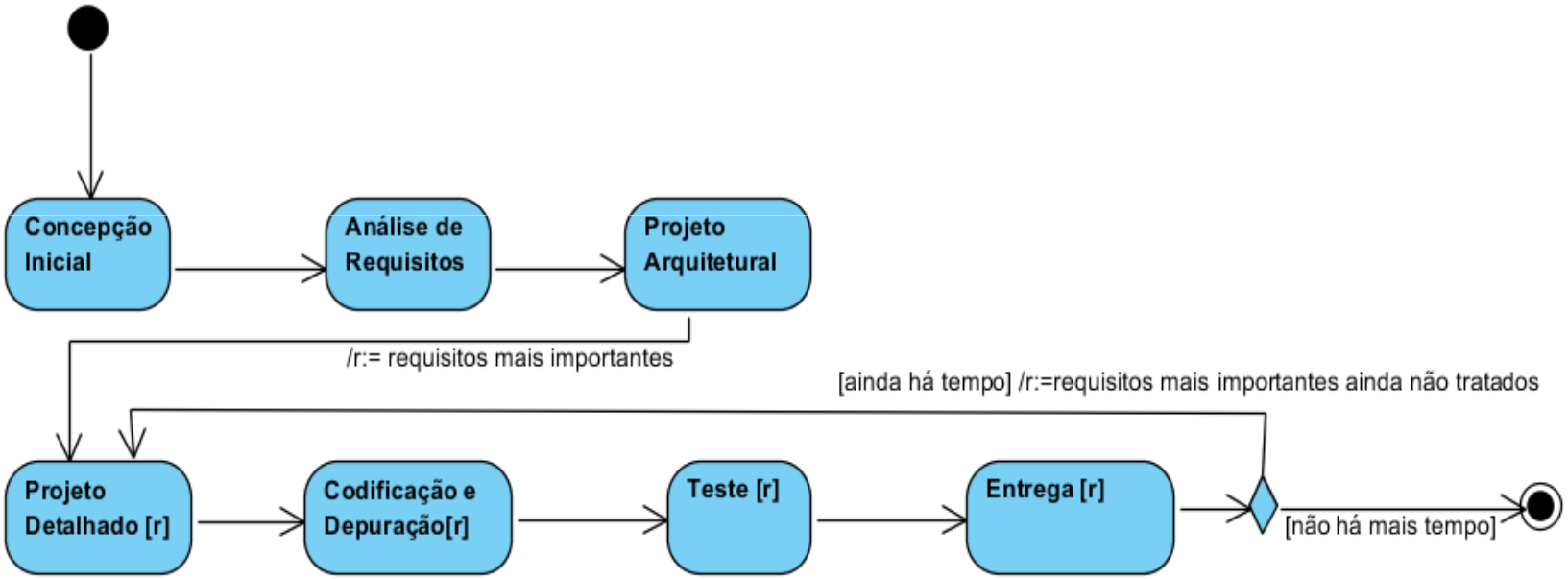
PROTOTIPAÇÃO EVOLUCIONÁRIA



Entregas em Estágios



Modelo Orientado a cronograma



Entrega Evolucionária

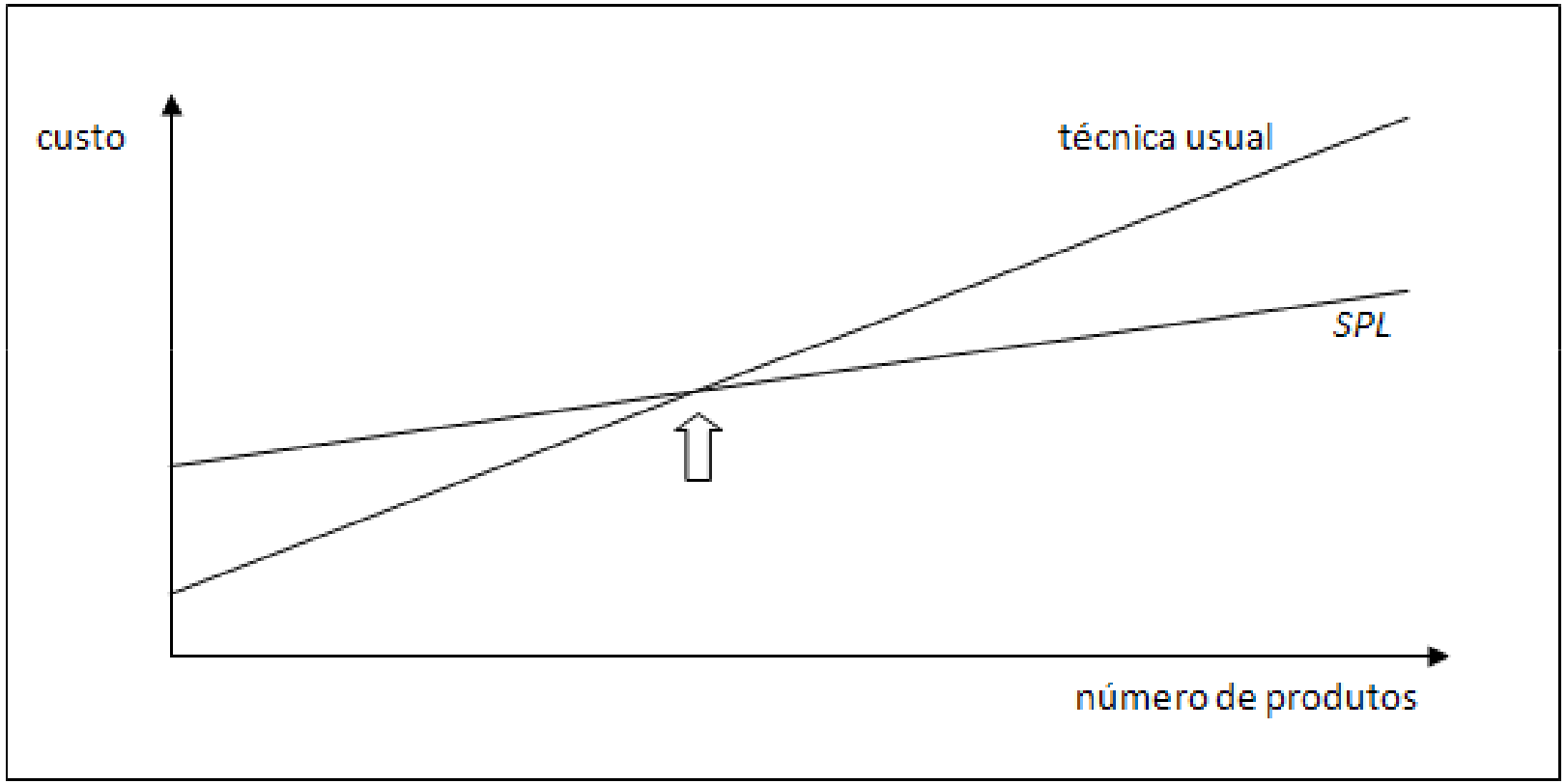
- Se a ideia é acomodar todos ou a grande maioria das modificações, então a abordagem tende mais para Prototipação Evolucionária.
- Se, entretanto, as entregas continuarão sendo planejadas de acordo com o previsto e as modificações acomodadas aos poucos nas entregas, então a abordagem se parece mais com Entrega em Estágios.

Modelos orientados a ferramentas

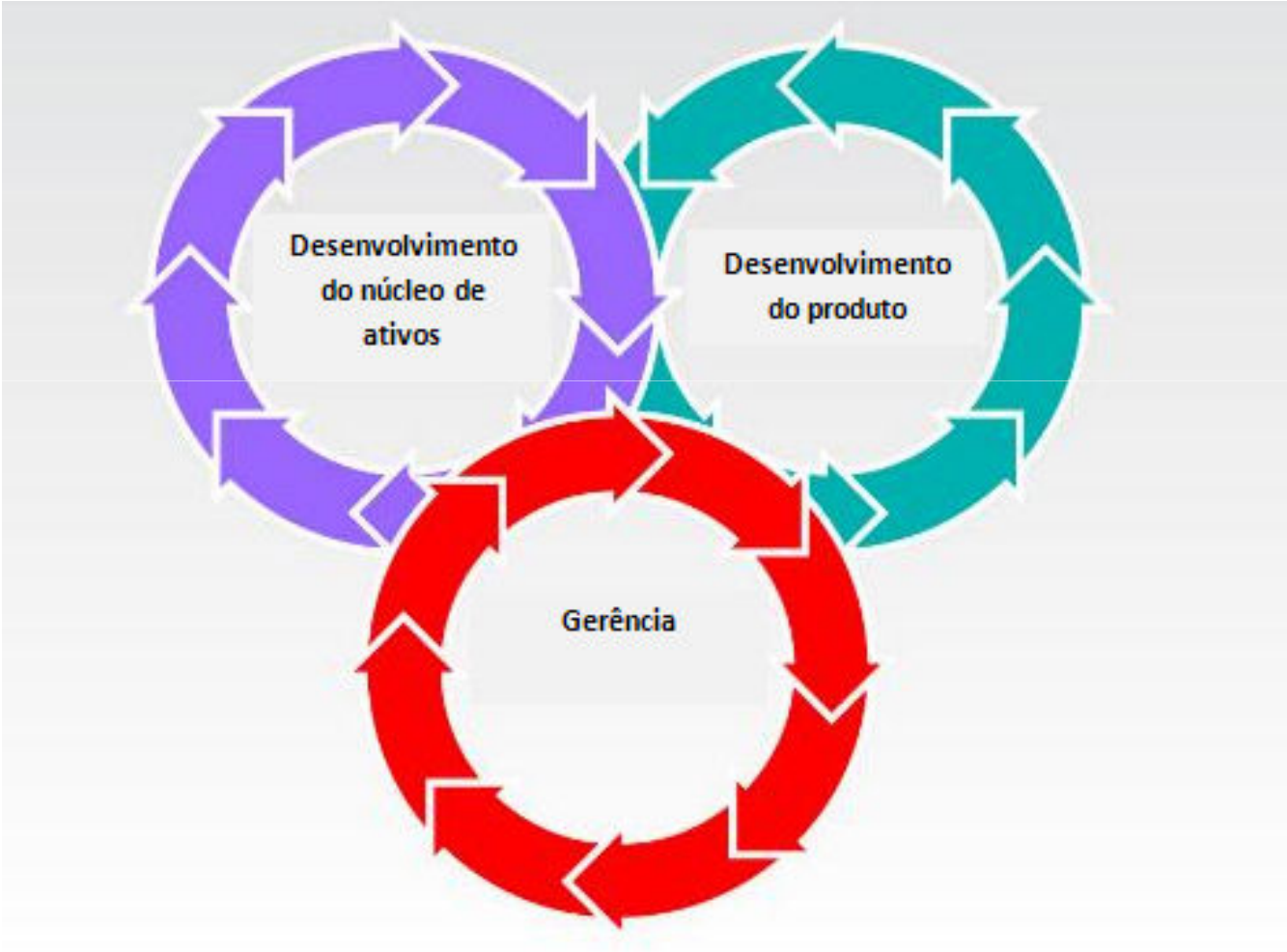
- Chama-se de *Modelo Orientado a Ferramentas* (*Design to Tools*) qualquer modelo baseado no uso intensivo de ferramentas de prototipação e geração de código, que permitem a rápida produção de sistemas executáveis a partir de especificações em alto nível.
- É uma abordagem extremamente rápida de desenvolvimento e prototipação, mas é limitada pelas funcionalidades oferecidas pelas ferramentas específicas.

Linhas de Produto de Software (SPL)

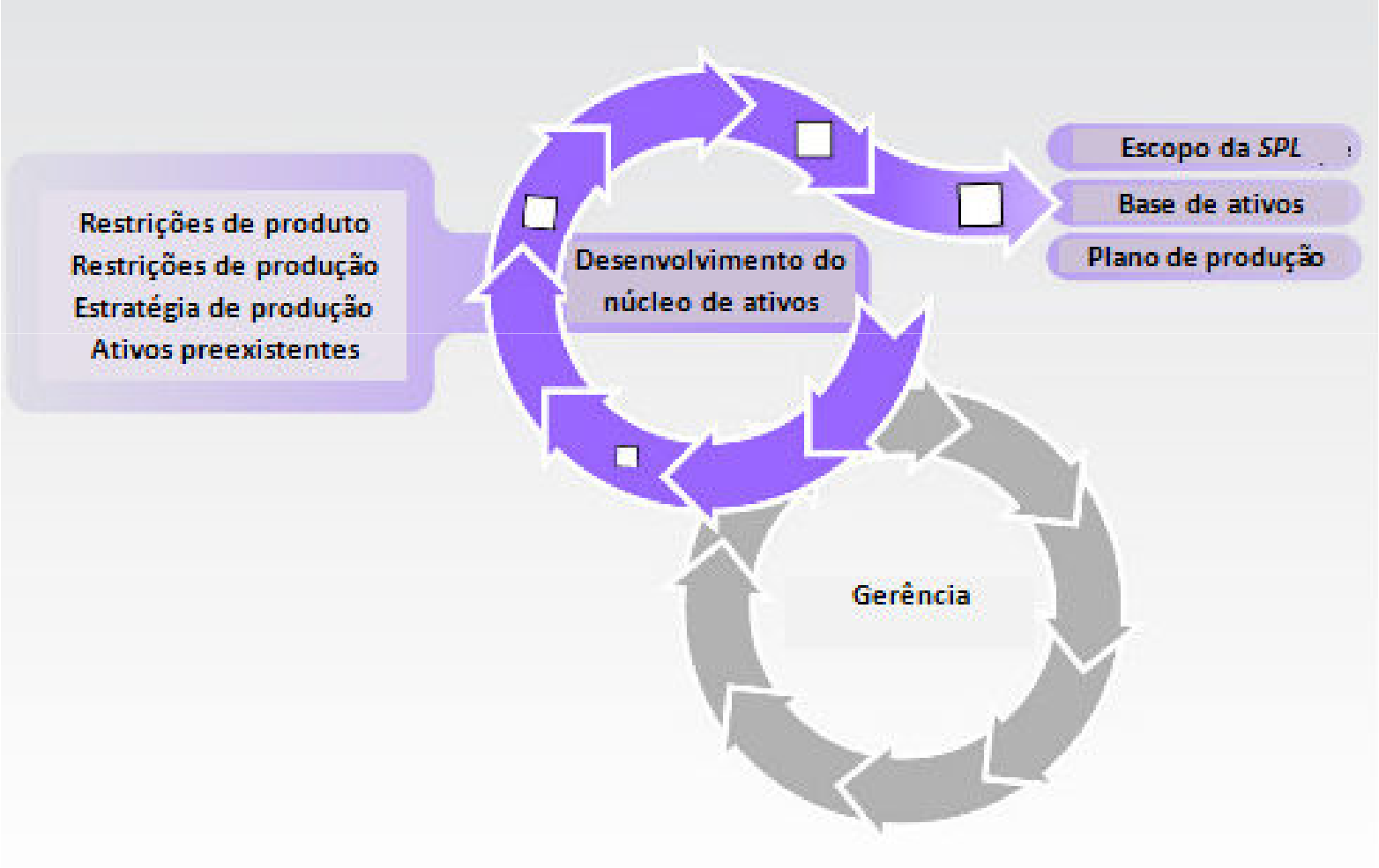
CUSTO/BENEFÍCIO DA SPL



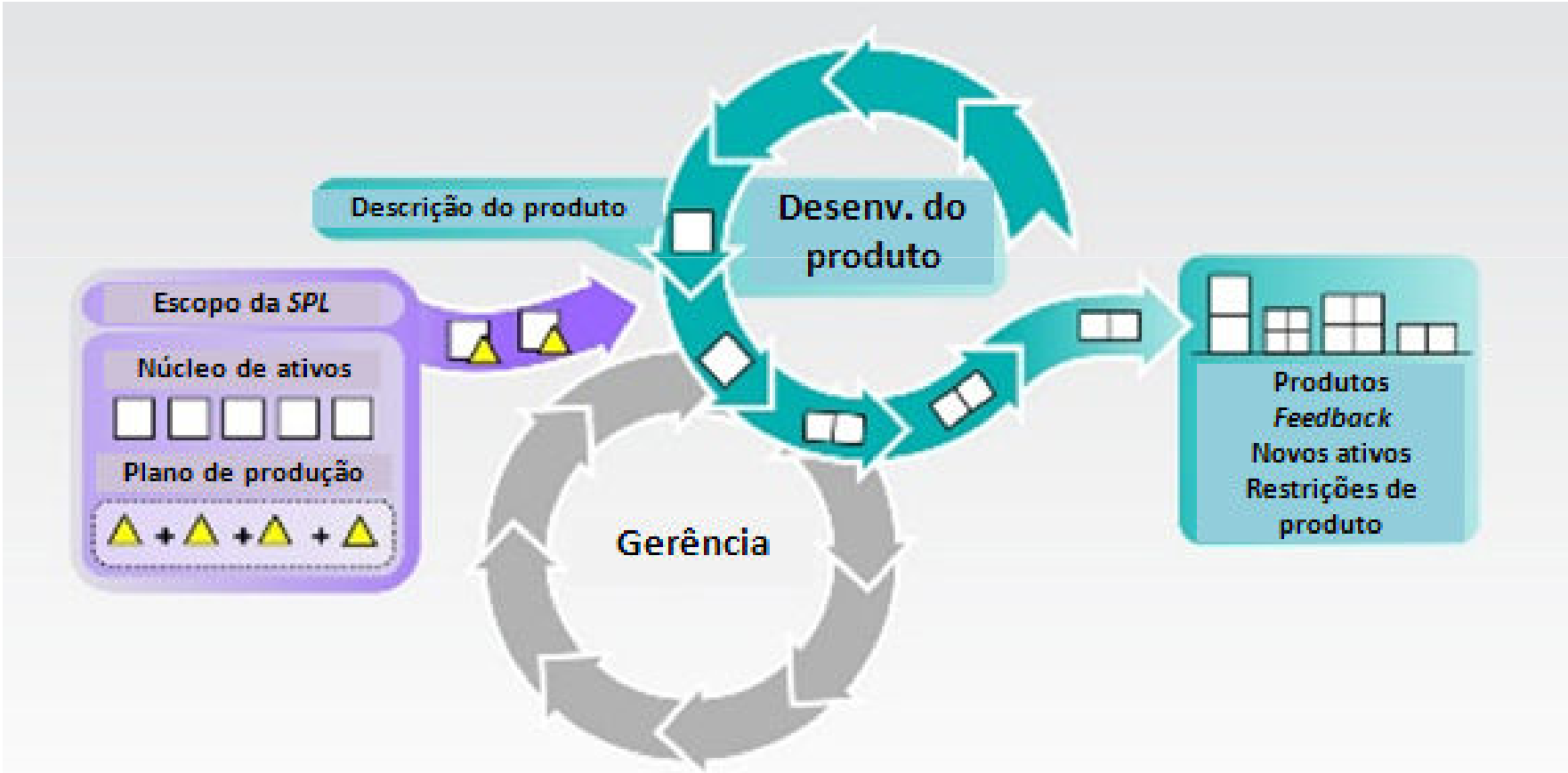
Atividades essenciais para SPL



Desenvolvimento do núcleo de ativos



Desenvolvimento do produto



Gerência

Gerência organizacional

- Identifica a estratégia de negócio e oportunidades com a *SPL*.
- Pode ser considerada como a responsável pelo sucesso ou fracasso de um projeto

Gerência técnica

- Deve acompanhar as atividades de desenvolvimento, verificando se os padrões são seguidos, se as atividades são executadas e se o processo pode ser melhorado