

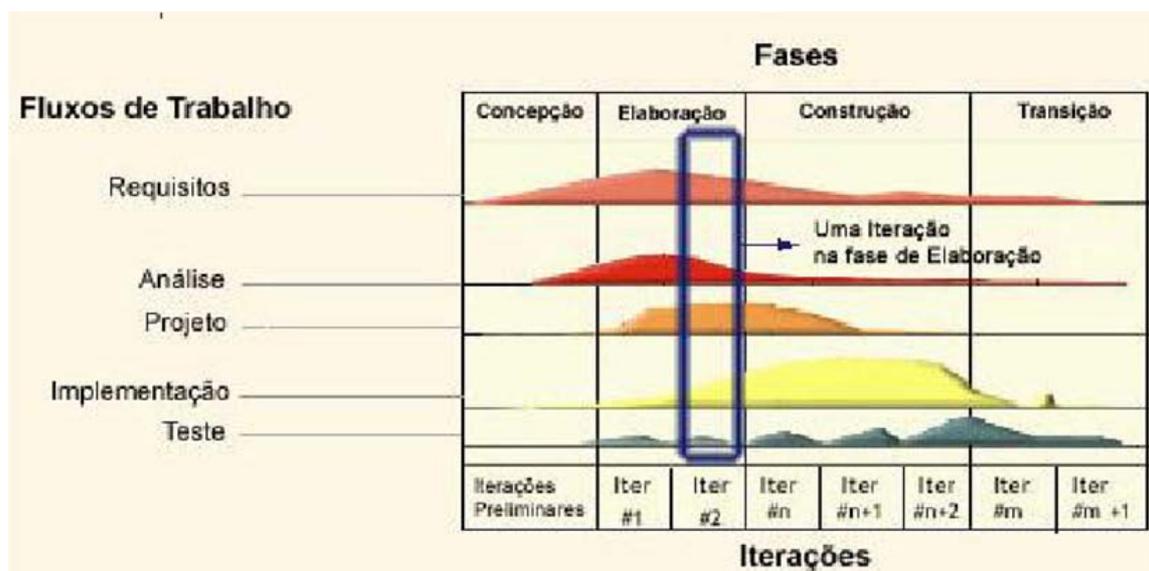
Processo Unificado

Universidade Federal do Maranhão – UFMA
Pós Graduação de Engenharia de Eletricidade
Grupo de Computação
Assunto: Ciclo de Vida - Fluxos

Autoria: Aristóфанes Corrêa Silva
Adaptação: Alexandre César M de Oliveira

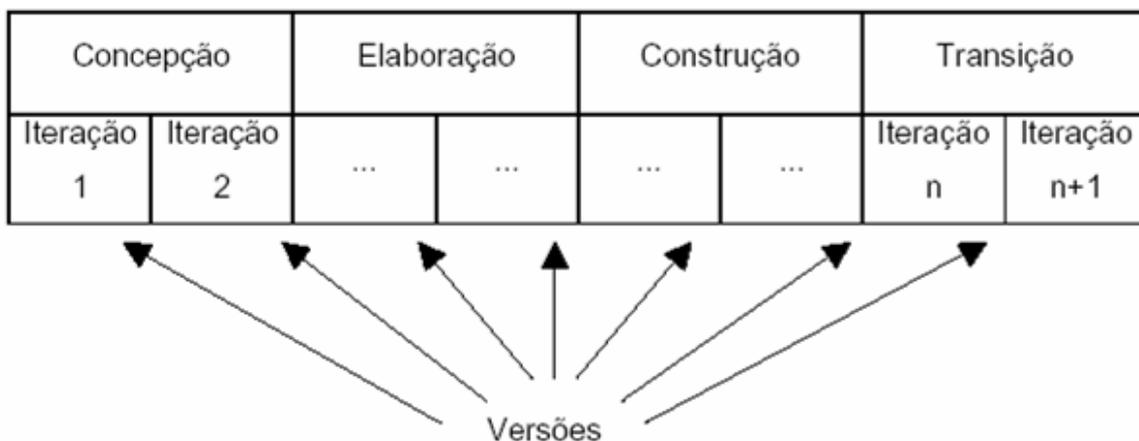
2 Fluxos no Ciclo de Vida do Processo Unificado

- O Processo Unificado consiste da repetição de uma série de ciclos durante a vida de um sistema.
- Cada ciclo é concluído com uma versão do produto pronta para distribuição e é subdividido em quatro fases: concepção, elaboração, construção e transição.
- Cada fase, por sua vez, é subdividido em iterações que passam por todos os cinco fluxos do trabalho do processo: análise de requisitos, análise, projeto, implementação e teste.



2.1 Interações

- Antes de sua conclusão, o ciclo de vida do processo unificado passa por várias e sucessivas iterações.
- Cada uma destas iterações resulta em uma versão de um produto executável que constitui um subconjunto do produto final em desenvolvimento e cresce de modo incremental de uma iteração para outra até se tornar o sistema final.

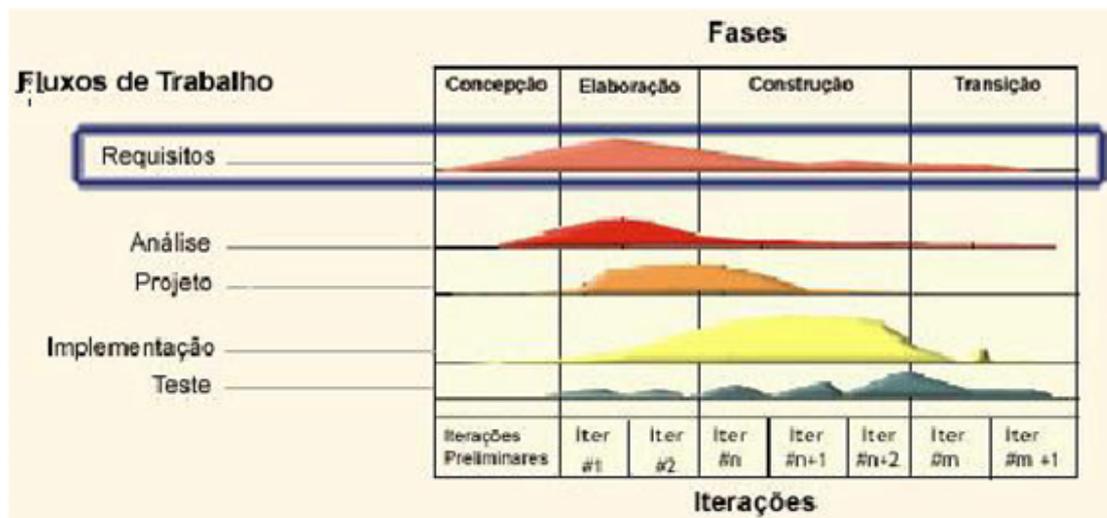


- Durante a concepção, o foco está na captação de requisitos.
- Durante a elaboração, o foco passa a ser a análise e projeto do sistema
- Durante a construção, a implementação é a atividade central
- E a transição é caracterizada pela entrega de um produto aos usuários

2.2 Fluxos de trabalho

- Uma iteração típica realiza cinco atividades ou fluxo de trabalho:
 - Requisitos
 - Análise
 - Projeto
 - Implementação
 - Teste

2.3 Requisitos

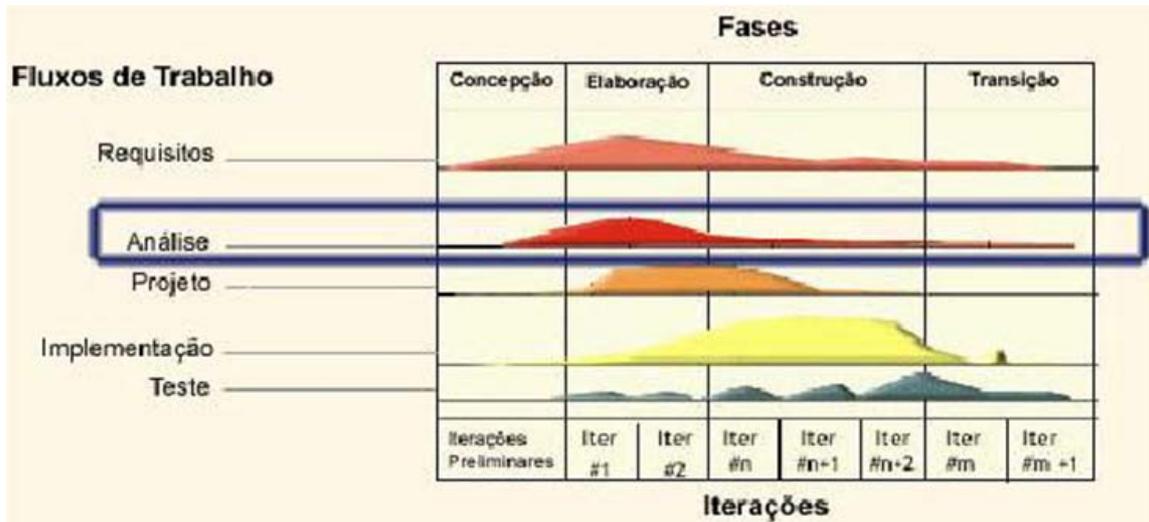


- Durante o fluxo de trabalho, os requisitos do sistema são especificados através da identificação das necessidades de usuários e clientes

- Estes requisitos funcionais são expressos em casos de uso através do modelo de casos de uso
- Um modelo de casos de uso é composto por todos os atores e casos de uso de um sistema, ou seja, é composto pelo conjunto de diagramas de casos de uso que compõem o sistema, e especifica como esse sistema será utilizado sob a perspectiva de clientes, usuários e desenvolvedores.
- O modelo de casos de uso é desenvolvido em vários incrementos, onde as iterações irão adicionar novos casos de uso e/ou adicionar detalhes as descrições dos casos de uso existentes.
- Durante a fase de concepção, os casos de uso mais importantes são identificados, delimitando o domínio do sistema.
- Durante a fase de elaboração, a maioria dos requisitos remanescentes é capturada, assim desenvolvedores poderão saber o quão deverão se empenhar para desenvolver o sistema
- Ao final da fase de elaboração, devem ter sido capturados e descritos cerca de 80% dos requisitos do sistema, porém, apenas 5% a 10% destes requisitos são implementados nesta fase
- Os requisitos que sobram são capturados e implementados durante a fase de construção.
- Na fase de transição quase não há requisitos a serem capturados, a menos que ocorram mudanças nos mesmos.

- O modelo de casos de uso estará concluído quando representar todos os requisitos funcionais do sistema de forma que usuários, clientes e desenvolvedores possam entender.
- O fluxo de trabalho de requisitos foca suas atividades na identificação d entidades que interagem com o sistema (atores) e dos requisitos funcionais deste sistema para cada um dos atores (casos de uso) e no agrupamento destes elementos sob determinado contexto de forma a construir diagramas de casos de uso que no seu conjunto formarão o modelo de caso de uso.
- O modelo de caso de uso é uma ferramenta utilizada para organizar os requisitos em forma fácil de gerenciar. Cliente e usuários devem entendê-lo e usá-lo para comunicar suas necessidades de forma consistente e não-redundante.
- Desenvolvedores podem dividir o trabalho de captura de requisitos entre si e, então, utilizar os resultados como entrada para os fluxos de análise, projeto, implementação e teste.

2.4 Análise

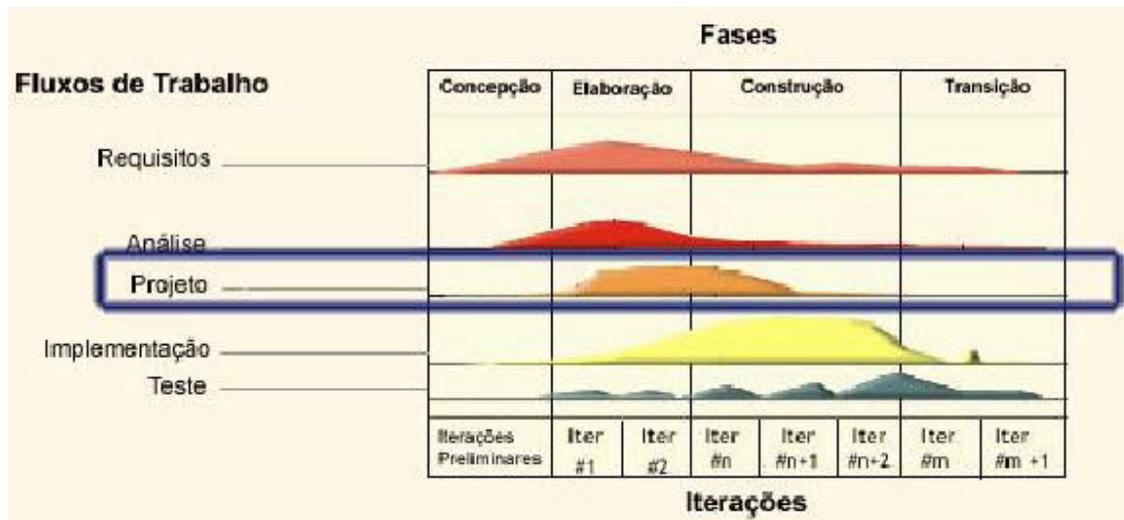


- O produto do fluxo de análise é o modelo de análise.
- O modelo tem a função de refinar os requisitos especificados no fluxo anterior (fluxo de requisitos) através da construção de diagramas de classes conceituais, permitindo, desta forma, argumentação a respeito do funcionamento interno do sistema.
- O modelo de análise fornece mais poder expressivo e formalismo, como diagramas de interações e diagrama de gráficos de estados que representam a dinâmica do sistema.
- O fluxo de análise tem maior importância durante fase de elaboração
- Isso contribui para a definição de uma arquitetura estável e facilita o entendimento detalhado dos requisitos.

- O modelo de análise cresce incrementalmente a medida em que casos de uso são analisados.
- Para cada iteração, é selecionado um conjunto de casos de uso que são realizados no modelo de análise.
- O sistema é construído como uma estrutura de classes de análise e relacionamentos entre estas classes.
- Na próxima iteração outro conjunto de casos de uso é realizado e adicionado ao produto da iteração prévia.
- A maneira para realizar este fluxo de trabalho é primeiro identificar e detalhar os casos de uso para uma iteração, e depois, através da análise da descrição de cada caso de uso, sugerir quais classes e relacionamentos são necessários para realizá-lo. Isto é feito para todos os casos de uso em uma iteração.
- Dependendo da etapa em que o ciclo de vida do processo se encontra e do tipo de iteração que está sendo trabalhada, a arquitetura do sistema poderá auxiliar a identificação de novas classes e a reutilização de outras já existentes.
- Cada classe de análise possui um ou vários papéis em uma descrição de um caso de uso.
- Cada papel especifica as responsabilidades, atributos, e tudo mais que for necessário para detalhar um caso de uso.
- Ao estruturar os requisitos do sistema, o modelo de análise fornece uma estrutura com foco na manutenção dos mesmos.

- Essa estrutura não é usada apenas à manutenção de requisitos, mas também serve de entrada para os fluxos de projeto e implementação
- O modelo de análise pode ser visto como o primeiro passo para o desenvolvimento do modelo de projeto.
- O fato de se preservar a estrutura do modelo de análise no projeto, permite que o processo desenvolva um sistema de fácil manutenção, que possua um grande poder de recuperação mediante a mudança de requisitos e que inclua elementos que possam ser reutilizados em sistemas relacionados em desenvolvimento.
- É bom salientar que o modelo de análise não trata problemas que são solucionados em modelos de projeto ou implementação.
- A estrutura fornecida pelo modelo de análise nem sempre deve ser preservada, mas sim transformada durante o projeto e implementação do sistema.
- A razão pela qual a preservação da estrutura de um modelo de análise não é efetuada na prática é evidenciada pelo fato do projeto considerar a plataforma de implementação do sistema como, linguagens de programação, SO, etc.
- Visando a diminuição de custos e o melhor aproveitamento de prazos, a arquitetura deve ser alcançada modificando-se o modelo de análise durante a transição para o modelo de projeto.

2.5 Projeto

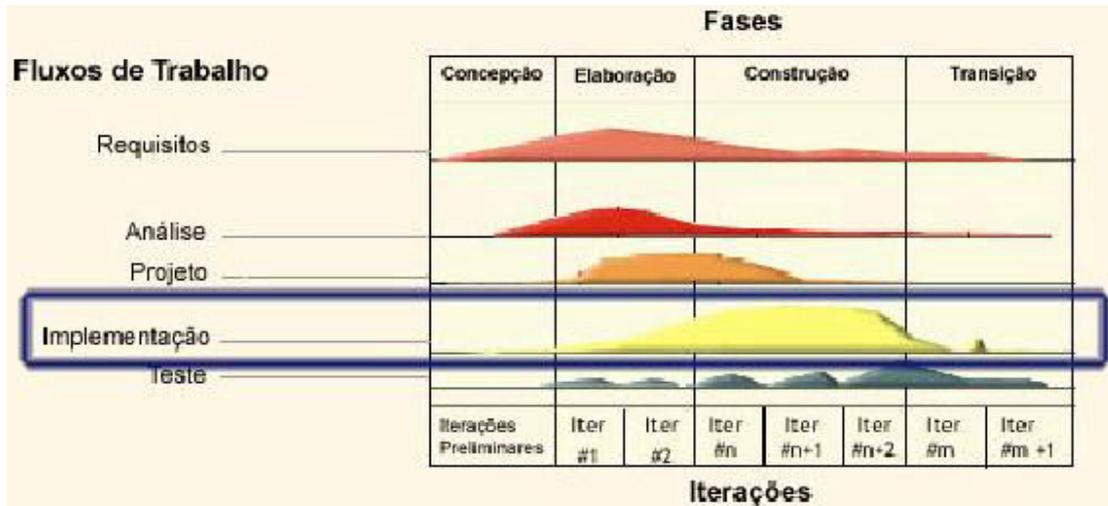


- No fluxo de projeto o sistema é moldado e sua forma é definida de maneira a suprir as necessidades especificadas pelos requisitos.
- Define um modelo de projeto que é construído com base no modelo de análise definido no fluxo anterior.
- Diferente do fluxo de análise, cujo produto é um modelo que descreve características comportamentais e estruturais do sistema em um nível conceitual, o fluxo de projeto desenvolve o modelo de projeto que descreve o sistema em um nível físico.
- A função principal deste fluxo é obter uma compreensão detalhada dos requisitos do sistema, levando em consideração fatores como linguagens de programação, SO, tecnologias de banco de dados, interface com o usuário, etc.
- O fluxo de projeto possui seu enfoque entre o fim da fase de elaboração e o início da fase de construção.

- Este fluxo serve de base para o fluxo de implementação
- Através do modelo de projeto, casos de uso são realizados por artefatos de projeto representados por ferramentas de modelagem também utilizadas no fluxo de análise como diagramas de classes, diagramas de interação e diagramas de estados, agora com o intuito de capturar os requisitos de implementação.
- Os mesmos diagramas citados são construídos em um nível mais físico que conceitual.
- O fluxo de projeto também pode utilizar o diagrama de objetos
- O modelo de projeto descreve as realizações físicas de casos de uso considerando como requisitos, e outros detalhes relacionados ao ambiente de implementação, causam impacto ao sistema sob consideração.
- O modelo de projeto funciona como uma abstração da implementação do sistema.
- Enquanto que o fluxo de análise se interessa por *o que* o sistema de fazer, o fluxo de projeto diz respeito a *como* os requisitos serão implementados e, portanto, pressupõe uma infra-estrutura de implementação e é fortemente influenciado por ela.
- O projeto pode dividir o sistema em subsistemas visando a organização dos artefatos do modelo de projeto e partes mais gerenciáveis

- Um subsistema pode ser composto de classes de projeto, realizações de caso de uso, interfaces, e outros subsistemas.
- Um subsistema pode fornecer interfaces que representam a funcionalidade que elas exportam em termos de operações.
- Um subsistema deve ser coesivo, isto é, seu conteúdo deve estar extremamente relacionado.
- Cada subsistema deve depender no mínimo de outro subsistema.
- A grande vantagem de se trabalhar com subsistemas no fluxo de projeto é que estes subsistemas poderão ser projetadas separadamente e concorrentemente por diferentes equipes de desenvolvimento em diferentes níveis do projeto.

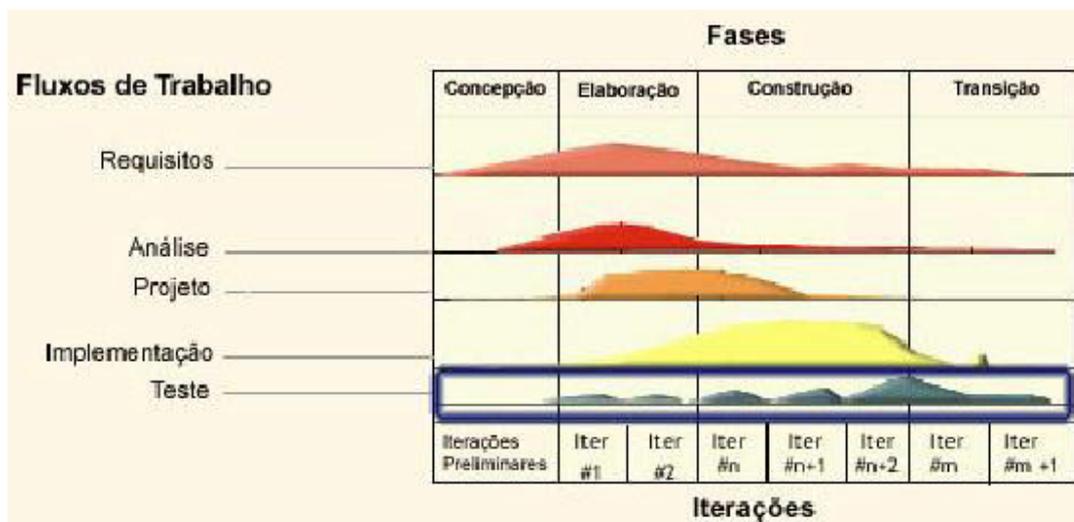
2.6 Implementação



- O fluxo de implementação é baseado no modelo de projeto
- Implementa o sistema em termos de componentes (código fonte, executável, etc).
- A maior parte da arquitetura do sistema é definida durante o projeto.
- O modelo de implementação que se limita:
 1. planejar as integrações do sistema em cada iteração. O resultado é um sistema que é implementado como uma sucessão de etapas pequenas e gerenciáveis
 2. implementar os subsistemas encontrados durante o projeto
 3. testar as implementações e integrá-los, compilando-as em um ou mais arquivos executáveis, antes de enviá-los ao fluxo de teste.

- Este fluxo tem maior importância na fase de construção e apesar de ter suas características próprias, a maior parte de suas atividades é realizada de forma quase mecânica, pelo fato das decisões mais difíceis terem sido tomadas durante o fluxo de projeto.
- O código gerado durante a implementação, deve ser uma simples tradução das decisões de projeto em uma linguagem específica.

2.7 Teste



- O fluxo de teste é desenvolvido com base no produto do fluxo de implementação.
- Os componentes executáveis são testados exaustivamente no fluxo de teste para então ser disponibilizados aos usuários finais.
- O principal propósito do fluxo de teste é realizar vários testes e sistematicamente analisar os resultados de cada teste.

- Componentes que possuem defeitos retornarão a fluxos anteriores como os fluxos de projeto e implementação, onde os problemas encontrados poderão ser corrigidos.
- O teste de um sistema é primeiramente empregado durante a fase de elaboração, quando a arquitetura do sistema é definida, e durante a fase de construção quando o sistema é implementado.
- Um planejamento inicial de testes pode ser feito durante a fase de concepção.
- Na fase de transição, o fluxo de testes se limita ao conserto de defeitos encontrados durante a utilização inicial do sistema.
- O produto do fluxo de teste é o modelo de teste, esse modelo primeiramente descreve como componentes executáveis, provenientes do fluxo de implementação, são testados.
- O modelo de testes também pode descrever como aspectos específicos do sistema testados, como por exemplo, se a interface do usuário é útil e consistente ou se o manual do usuário cumpre seu objetivo.
- O papel do fluxo de teste é verificar se os resultados do fluxo de implementação cumprem os requisitos estipulados por clientes e usuários, para que possa ser decidido se o sistema necessita de revisões ou se o processo de desenvolvimento pode continuar.