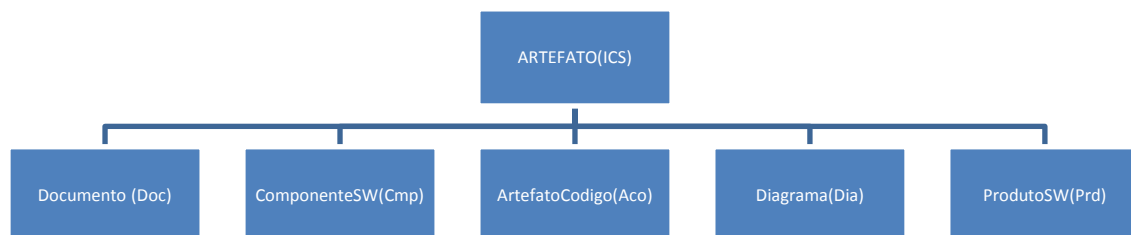




HIERARQUIA DE ARTEFATO OU ITEM DE CONFIGURAÇÃO (ICS)



DESCRIÇÃO

Item de Configuração de Software (ICS) item que está sob gerência de configuração e, assim, só pode ser alterado segundo um procedimento de controle de alteração formalmente estabelecido e documentado. Pode possuir várias variações. Pode ser uma ferramenta de software ou um artefato, como, por exemplo, um determinado plano de projeto ou certo artefato de código.

Artefatos (ICS) são insumos para, ou um produto de, uma atividade, no sentido de ser um objeto de transformação da atividade. Em função de sua natureza, artefatos podem ser classificados em: documentos, diagramas, artefatos de código, componentes de software ou produtos de software.



Documentos (Doc) são artefatos de software não passíveis de execução, constituídos tipicamente de declarações textuais, normalmente associados a padrões organizacionais (roteiros) que definem a forma como eles devem ser produzidos. Exemplos de documentos incluem: documento de especificação de requisitos, plano de projeto, plano de qualidade, relatório de avaliação da qualidade, entre outros.

Diagramas (Dia) são artefatos gráficos que consistem em apresentações gráficas de um conjunto de elementos de modelo, em geral representadas como um gráfico conectado de vértices (elementos de modelo) e arcos (relacionamentos). Exemplos de diagramas incluem: diagrama de casos de uso, diagrama de classes, diagrama de entidades e relacionamentos, diagrama relacional, entre outros.

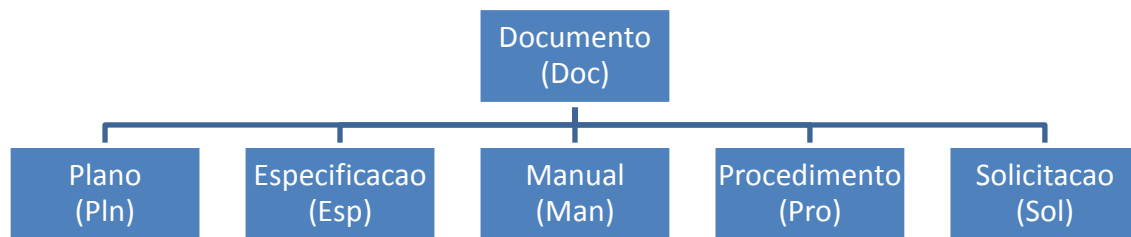
Artefatos de Código (Aco) são porções de código, passíveis de execução, geradas no próprio desenvolvimento. Exemplos de artefatos de código incluem: programas, subprogramas, classes, rotinas, funções, métodos, entre outros.

Componentes de Software (Cmp) são artefatos de software que provêm um conjunto bem definido de serviços, desenvolvidos de forma independente e manipulados (usados, vendidos, mantidos em uma biblioteca etc.) como unidades coerentes, que podem ser combinados para formar artefatos maiores, fornecendo uma interface bem definida e explícita. Exemplos de componentes de software incluem: classes, frameworks, padrões gerativos, entre outros.

Produtos de Software (Prd) são os artefatos resultantes de um processo de desenvolvimento de software concluído e entregues ao usuário, ou seja, colocados em operação. Incluem o conjunto de todos os artefatos gerados no desenvolvimento, além das ferramentas e softwares necessários para colocar os produtos em operação.



HIERARQUIA DE DOCUMENTO (Doc)



Documento (Doc) é o artefato de software não passível de execução, constituído tipicamente de declarações textuais, normalmente associados a padrões organizacionais (roteiros) que definem a forma como ele deve ser produzido. Exemplos de documentos incluem: documento de especificação de requisitos, plano de projeto, plano de qualidade, relatório de avaliação da qualidade, entre outros.

Plano (Pln) de software é um documento que especifica, em linguagem escrita, um processo de planejamento que se quer iniciar ou que já está iniciado.

Especificação (Esp) de software é o documento que contém instruções para o entendimento de um item, processo ou condição de software. É semelhante ao o plano, mas com menos detalhes e profundidade e refere-se a normas e decisões inerentes ao tipo de negócio.

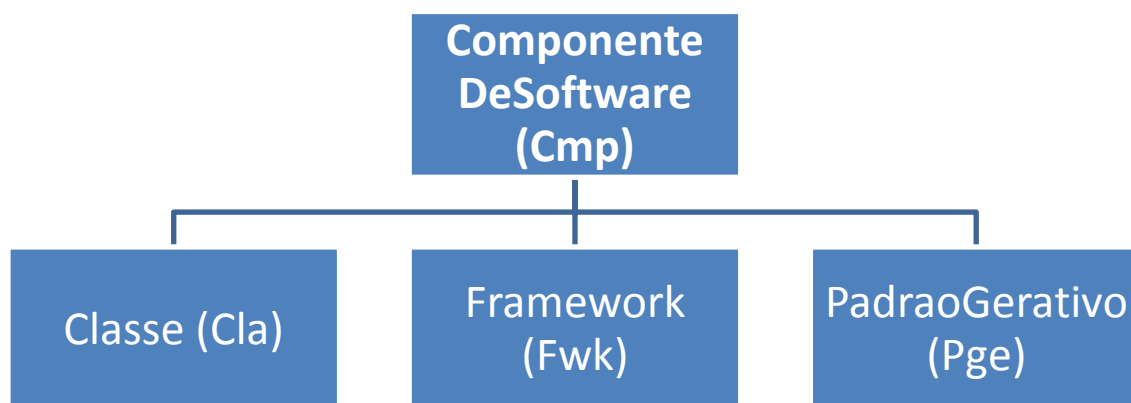
Manual (Man) de software pode ser um livro, folheto ou arquivo informático que ensina a operá-lo. Muitas vezes o manual vem com imagens, para não só ilustrá-lo, como ajudar na compreensão.



Procedimento (Pro) é uma descrição detalhada de todas as operações necessárias para a realização de uma atividade, ou seja, é um roteiro padronizado para realizar uma atividade. Ex: o check-list de testes é um procedimento de checagem dos passos de teste.

Solicitação (Sol) é o ato de enviar mensagens para pedido de mudanças, adições, exclusões e outras mudanças ao projeto de software. Ex: Pedido de Mudança.

HIERARQUIA DE COMPONENTE DE SOFTWARE (Cmp)



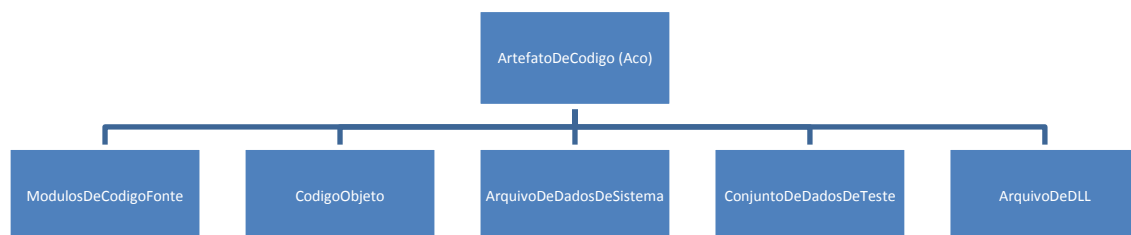
Componentes de Software (Cmp) são artefatos de software que provêem um conjunto bem definido de serviços, desenvolvidos de forma independente e manipulados (usados, vendidos, mantidos em uma biblioteca etc.) como unidades coerentes, que podem ser combinados para formar artefatos maiores, fornecendo uma interface bem definida e explícita. Exemplos de componentes de software incluem: classes, frameworks, padrões gerativos, entre outros.



Classe (Cla) representa um conjunto de objetos com características afins. Uma classe define o comportamento dos objetos através de seus métodos, e quais estados ele é capaz de manter através de seus atributos.

Framework (Fwk) ou **arcabouço**, em desenvolvimento de software, é uma abstração que une códigos comuns entre vários projetos de software provendo uma funcionalidade genérica. Um *framework* pode atingir uma funcionalidade específica, por configuração, durante a programação de uma aplicação. Ao contrário das bibliotecas, é o *framework* quem dita o fluxo de controle da aplicação, chamado de Inversão de Controle.

HIERARQUIA DE ARTEFATO DE CÓDIGO (Aco)



Artefatos de Código (Aco) são porções de código, passíveis de execução, gerados no próprio desenvolvimento. Exemplos de artefatos de código incluem: programas, subprogramas, classes, rotinas, funções, métodos, entre outros.



Módulos de Código Fonte são porções de código-fonte que, junto a outros artefatos de código, constituem o código executável.

Código-Objeto é o nome dado ao código resultante da compilação do código fonte. Normalmente, o código fonte é formado por uma sequência de instruções no formato especificado por alguma linguagem de programação. Para cada arquivo de código fonte é gerado um arquivo com código objeto, que posteriormente é "ligado" aos outros, através de um linker, resultando num arquivo executável ou biblioteca.

Arquivo de Dados de Sistema é um arquivo onde os dados do sistema são guardados para utilização, análise ou qualquer finalidade que se faça necessário o acesso a estes dados.

Conjunto de Dados de Teste - um conjunto de valores de entrada de teste que são usados durante a execução de um teste, e os resultados esperados mencionados para fins de comparação durante a execução de um teste.

Arquivo de DLL (Data Link Library) - É um arquivo que contém, um conjunto de funções e rotinas de programação que podem ser acessadas dinamicamente por um programa.



HIERARQUIA DE DIAGRAMA (Dia)

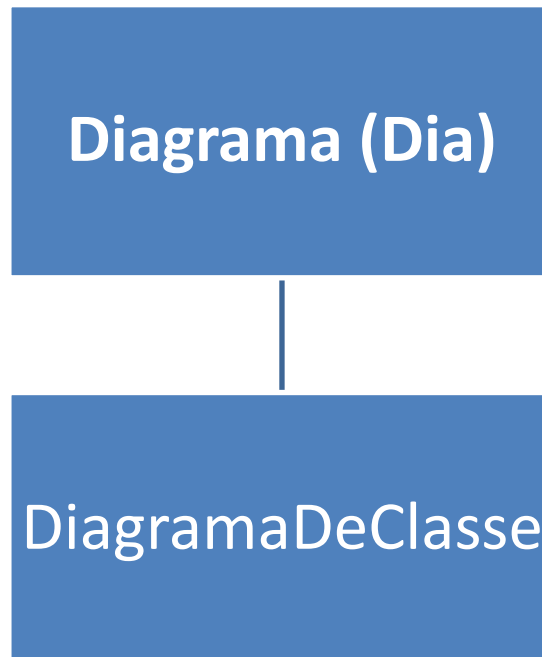
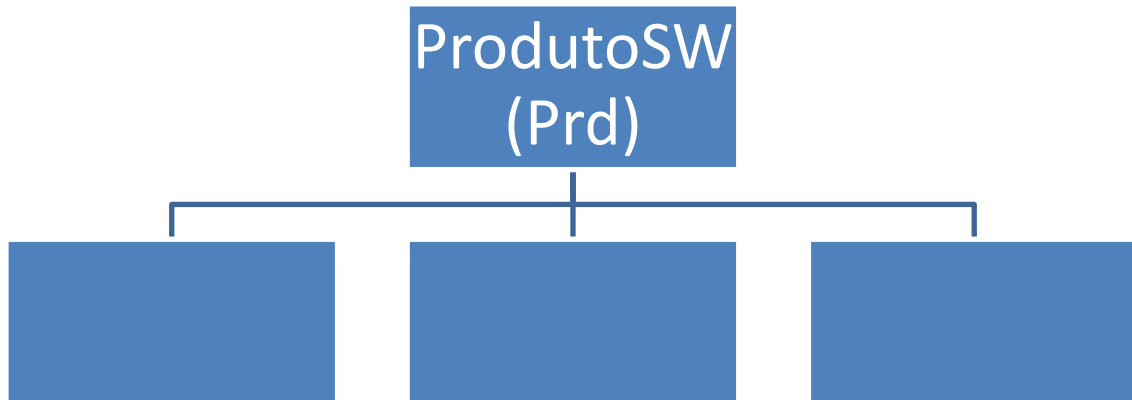


Diagrama (Dia) - Artefato gráfico, não passível de execução, que consiste em uma apresentação gráfica de um conjunto de **elementos de modelo**, em geral representada como um gráfico conectado de vértices (**elementos de modelo**) e arcos (**relacionamentos**). Ex.: diagrama de casos de uso, diagrama de classes, diagrama de entidades e relacionamentos, diagrama relacional etc.

Diagrama de Classe - é uma representação da estrutura e relações das classes que servem de modelo para objetos. É uma modelagem muito útil para o sistema, define todas as classes que o sistema necessita possuir e é a base para a construção dos diagramas de comunicação, sequência e estados.



HIERARQUIA DE PRODUTO DE SOFTWARE (Prd)



Produto de Software (Prd) - Artefato resultante de um processo de desenvolvimento de software concluído e entregue ao usuário, ou seja, colocado em operação. Inclui o conjunto de todos os artefatos gerados no desenvolvimento, além das ferramentas e softwares necessários para colocar o produto em operação. Em um ciclo de vida sequencial, qualquer alteração necessária em um produto de software se dará em uma atividade de manutenção e não mais em uma atividade de desenvolvimento do software. Em um ciclo de vida iterativo, porém, alterações podem significar a criação de variações de um produto de software.

Emerson José Porfírio