

Vamos entender construção de software?

CONSTRUÇÃO DE SOFTWARE - INSTITUTO DE INFORMÁTICA - UFG

Copyright © 2019

Fábio Nogueira de Lucena

Segundo a norma ISO/IEC/IEEE 12207:2017, um sistema de software inclui hardware, software, dados, seres humanos, processos, procedimentos, serviços, materiais e outros elementos que, em conjunto, são vistos pelos usuários como produto ou serviço. De acordo com esta norma um sistema de software é um sistema no qual o elemento software é de grande relevância para os interessados (*stakeholders*). Ou seja, em um sistema de software, o software é o principal elemento. Convém esclarecer que software não é capaz de ser executado sem um hardware e, portanto, sempre será parte de um sistema maior que o engloba.

O foco de interesse deste documento é o elemento software de sistemas de software. Apesar disso, não se pode ignorar os demais elementos, nem a relação deles com software.

O software de um sistema de software, doravante apenas software, por simplicidade, é um produto feito pela ação intencional do homem. Não ocorre por acaso, ao contrário, decorre de esforço significativo e articulado, despendido por profissionais com várias habilidades. Esse esforço está organizado em processos de software. E para executá-los é necessário habilidades, desenvolvidas, dentre outras possibilidades, em cursos de graduação como o curso de Bacharelado em Engenharia de Software da UFG.

Em tempo, Engenharia de Software é o nome atribuído à área que reúne o conhecimento pertinente ao desenvolvimento de software.

A norma ISO/IEC/IEEE 12207:2017 identifica dezenas de processos de software e, dentre eles, aquele pertinente à construção de software, denominado de *processo de implementação* (nosso foco de interesse).

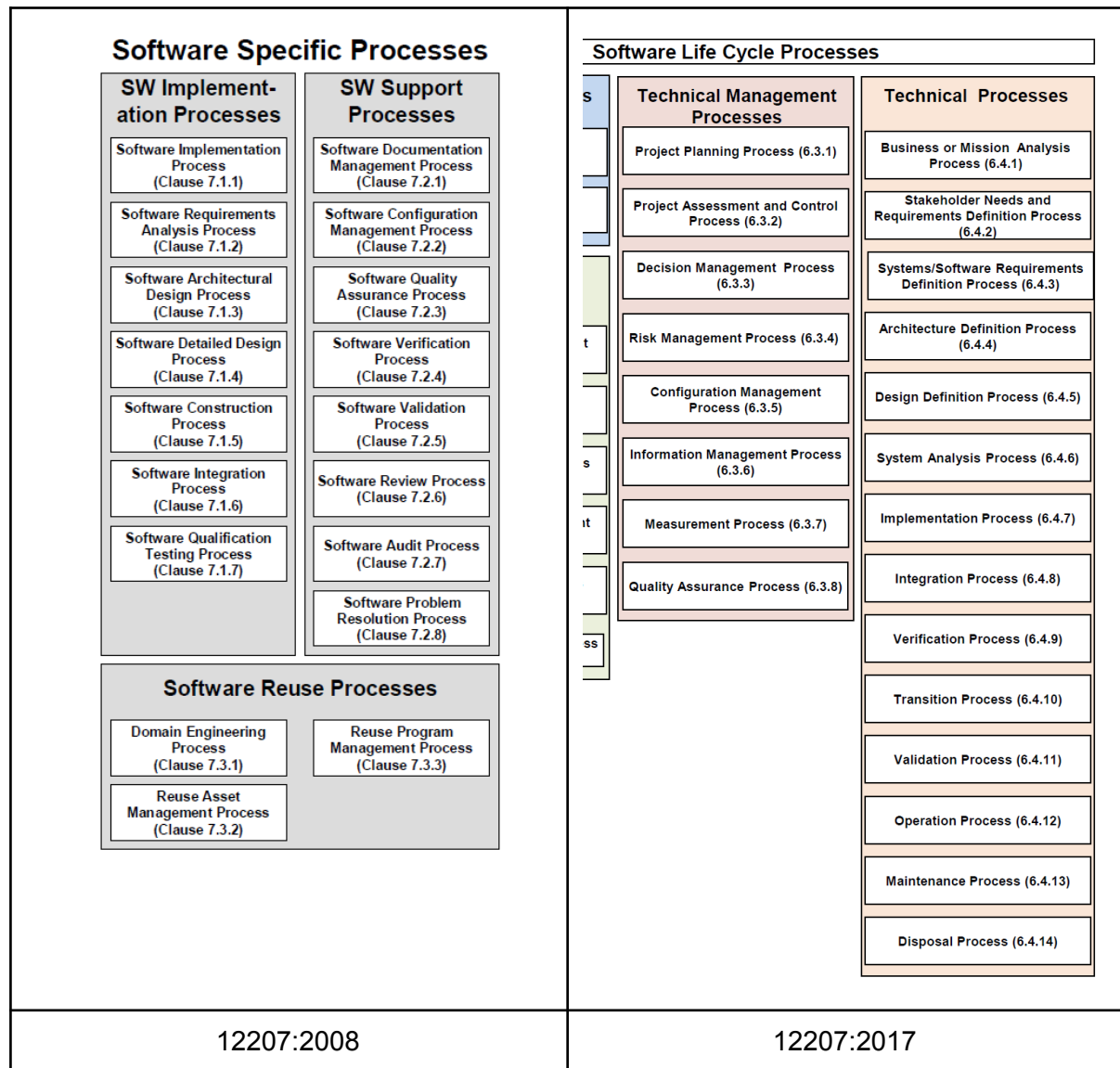
O que é construção de software?

Segundo a norma ISO/IEC/IEEE 24765:2017, construção é a atividade de desenvolvimento de software que compreende projeto detalhado, codificação, testes de unidade e depuração (*debugging*).

Esta norma também define implementação como o processo que traduz o projeto (*design*) em componentes de hardware, componentes de software ou ambos, ou ainda, o resultado desse processo. A norma também define implementação com a mesma acepção de construção.

O uso análogo destes dois termos, construção e implementação, também pode ser notado na atualização da norma ISO/IEC/IEEE 12207:2008, que empregava o termo construção, e cuja versão mais recente, ISO/IEC/IEEE 12207:2017, emprega o termo implementação com a mesma acepção. Em outras palavras, o que antes era Processo de Construção de Software (descrito na seção 7.1.5 daquele padrão) agora é Processo de Implementação (descrito na seção 6.4.7 do padrão atualizado). A figura abaixo ilustra excertos de ambas as normas com o propósito de facilitar a identificação desta mudança.

Fonte: ISO/IEC/IEEE 12207:2008 e ISO/IEC/IEEE 12207:2017.



Na ISO/IEC/IEEE 12207:2017, o processo de implementação é um dos processos técnicos. Os processos técnicos são aqueles que transformam as necessidades dos interessados (*stakeholders*) em um produto ou serviço. Contudo, convém ressaltar, a definição das necessidades dos interessados também é um dos processos técnicos (conforme pode ser constatado na figura acima, lado direito).

O processo de implementação tem como objetivo “produzir um elemento de um sistema de software”. Noutras palavras, transformar requisitos, arquitetura, e projeto (*design*), incluindo interfaces, por meio de ações, em um elemento de um sistema de software em conformidade com as práticas tecnológicas de implementação selecionadas, usando disciplinas ou especialidades técnicas apropriadas.

O resultado do processo de implementação é um elemento de um sistema de software que satisfaz os requisitos especificados (inclusive os requisitos alocados¹ e derivados²), a arquitetura e o projeto (*software design*). Este resultado também é conhecido por "item de software".

Na perspectiva da ISO/IEC/IEEE 12207:2017, arquitetura e projeto (*software design*) não se confundem. A arquitetura define modelos e visões de uma solução que satisfaz as necessidades e os requisitos dos interessados, enquanto o projeto (*software design*) acrescenta detalhes necessários à arquitetura, ou seja, aos modelos e às visões definidos, com o propósito de alimentar adequadamente o processo de implementação.

Por fim, convém fornecer a definição de "construção de software" segundo o SWEBOK (*Software Engineering Guide of Knowledge*, V3), no qual "construção de software refere-se à criação de software por meio da combinação de codificação, verificação, testes de unidade, testes de integração e depuração (*debugging*)."

Processo de implementação

Resultados

Os resultados produzidos pela execução satisfatória do processo de implementação são:

- a. Restrições de implementação que influenciam os requisitos, arquitetura, ou projeto (*software design*) são identificadas. *Explicação.* Ao longo da implementação pode ser detectada a necessidade de um requisito adicional para o correto funcionamento da solução conforme esperado; ou a necessidade de ajuste de um componente da arquitetura para efetivamente viabilizar o seu uso pela implementação em andamento.
- b. Um elemento do sistema de software é produzido. *Explicação.* Este é o resultado mais claro do processo de implementação. O que foi criado pela construção, contudo, não pode ser interpretado como o único resultado, conforme os demais itens desta lista demonstram.

¹ Segundo a ISO/IEC/IEEE 24765:2017, um requisito alocado é um requisito ou parte de um requisito atribuído a um elemento arquitetural de mais baixo nível ou componente de projeto. Ou seja, um requisito alocado pode ser obtido pela divisão ou refinamento de um requisito existente.

² Segundo a ISO/IEC/IEEE 24765:2017, um requisito derivado é um requisito inferido ou deduzido da coleção de requisitos de um dado sistema. Ou seja, trata-se de um requisito implícito pelo contexto.

- c. Um elemento do sistema de software é empacotado ou armazenado. *Explicação.* O que foi produzido precisa ser preparado para a implantação, no caso de software feito em Java, por exemplo, pode ser que um arquivo .jar seja criado, ou que o que foi produzido simplesmente seja disponibilizado para o consumo por outros.
- d. Qualquer sistema ou serviço habilitador (*enabling system*), necessário para a implementação, é disponibilizado. *Explicação.* A implementação pode depender de vários serviços, por exemplo, para controle de versão e integração contínua, dentre outros e, neste caso, os serviços correspondentes precisam ser disponibilizados. Observe que não se está falando de um sistema ou serviço empregado durante a operação do software em construção, ou seja, um sistema ou serviço que contribui direta ou indiretamente com as funções do software em construção.
- e. Rastreabilidade é estabelecida. *Explicação.* A rastreabilidade bidirecional é mantida entre o elemento implementado e a correspondente arquitetura de software, projeto (*software design*) e a especificação de requisitos. Também é estabelecida a rastreabilidade para requisitos correlatos como planos de validação e verificação, dentre outros.

Atividades e tarefas

O processo de implementação é dividido em várias atividades classificadas em três grupos: (a) preparar para a implementação; (b) executar a implementação e (c) gerenciar os resultados da implementação. Cada um destes grupos é apresentado abaixo, juntamente com as atividades correspondentes. Tais atividades e tarefas devem ser implementadas em conformidade com as políticas e procedimentos organizacionais aplicáveis e pertinentes ao processo de implementação.

Preparar para a implementação

1. Definir uma estratégia de implementação, geralmente registrada no Plano de Desenvolvimento de Software e algumas vezes no Plano de Gerenciamento de Projeto. Tal estratégia deve considerar:
 - i. políticas e padrões de desenvolvimento, inclusive padrões aplicáveis de segurança, privacidade e práticas ambientais; padrões de programação ou codificação; políticas de teste de unidade; e padrões específicos por linguagem para implementação de recursos de segurança;
 - ii. Para software a ser reutilizado ou adaptado, métodos para determinar o nível, fonte, e adequação dos elementos de sistema a serem reutilizados e a segurança da cadeia de fornecimento (*supply chain*);
 - iii. procedimentos e métodos para o desenvolvimento de software (construção) e desenvolvimento de testes de unidade; além do uso de revisão por pares, testes de unidade e inspeções (*walkthrough*) durante a implementação;

- iv. uso de gerenciamento de configuração³ (*Configuration Management*) durante a construção de software;
 - v. considerações de gerenciamento de mudança para processos manuais;
 - vi. prioridades de implementação para apoiar a migração e a transição de dados e software, juntamente com o fim de operação (aposentadoria) de sistemas legados;
 - vii. criação de procedimentos de teste automatizados ou manuais para verificar que uma unidade de software satisfaz seus requisitos antes da criação da unidade de software (*test-driven development*); e
 - viii. ambientes especializados de suporte e desenvolvimento do ciclo de vida para realizar e gerenciar requisitos, modelos e protótipos, elementos entregáveis de software ou de sistema, além de especificações de testes e casos de teste.
2. Identificar restrições da estratégia de implementação e das tecnologias de implementação nos requisitos do software, nas características da arquitetura, nas características do projeto ou nas técnicas de implementação.
- i. Restrições incluem limitações correntes ou antecipadas da tecnologia de implementação escolhida (por exemplo, para software, o sistema operacional, o sistema de gerenciamento de banco de dados e serviços web), bem como aquelas resultantes do uso de sistemas habilitadores (sistemas que fornecem serviços essenciais).
3. Identificar e criar planos para os ambientes de software necessários, incluindo os sistemas habilitadores ou serviços exigidos pelo desenvolvimento e testes.
- i. A implementação de software geralmente faz uso de ambientes específicos e distintos daquele operacional (ou de produção). As configurações destes ambientes são controladas. O processo de implementação, sistemas habilitadores e serviços comuns incluem a demanda por ambientes de suporte e desenvolvimento do ciclo de vida especializados para realizar e gerenciar requisitos, modelos e protótipos, elementos entregáveis, ambientes de teste, especificações e casos de teste; simuladores para sistemas externos, sistemas de treinamento; e sistemas de gerenciamento de conteúdo para documentação do usuário.
4. Obter ou adquirir acesso aos ambientes de software e outros sistemas habilitadores ou serviços.

Realizar a implementação

1. Produzir ou adaptar elementos de software de acordo com a estratégia, restrições e procedimentos de implementação definidos.
- i. Elementos de software são adquiridos, identificados para reutilização de ativos organizacionais, ou desenvolvidos (construídos). Elementos de

³ Segundo Ian Sommerville em *Software Engineering*, 10th Edition, Pearson, 2016, “gerenciamento de configuração é o processo de gerenciar mudanças em um sistema de software e inclui quatro atividades fundamentais: (a) controle de versão (Git, por exemplo); (b) integração de sistema (GNU Build System, por exemplo); (c) acompanhamento de erros (Bugzilla, por exemplo) e (d) gerenciamento de liberações (*release management*) onde software é preparado para liberação externa.

software que são adquiridos podem variar de um simples produto comprado em conformidade com as regras de compra do projeto ou da organização em questão até aquisições complexas de um sistema de software que envolve processos de aquisição e suporte. A adaptação inclui a configuração de elementos de software que são reutilizados ou modificados. A construção pode envolver a codificação, reutilização adaptada e integração de unidades existentes, refatoração, desenvolvimento de bases de dados e a construção manual ou automatizada de procedimentos de teste para cada unidade.

- ii. Para elementos de software que são desenvolvidos, unidades de software executáveis são construídas (em geral com as estruturas de dados associadas, interfaces de programação de aplicação, descrições de serviço, documentação do usuário, casos de teste ou outros elementos), controladas, disponibilizadas aos papéis autorizados e armazenadas conforme os procedimentos de gerenciamento de configuração para artefatos de desenvolvimento.
2. Adquirir ou adaptar elementos de hardware do sistema de software.
 - i. Elementos de hardware são adquiridos ou fabricados usando técnicas aplicáveis tendo em vista a tecnologia de implementação física e materiais selecionados. Quando apropriado, a conformidade de elementos de hardware é verificada em relação aos requisitos especificados do sistema e características críticas de qualidade. Nos casos de produção em massa, os procedimentos de implementação e processos de fabricação são definidos e podem ser automatizados visando a repetição e consistência do que é produzido. Alguns elementos de hardware comuns em sistemas de software incluem integrações de sistemas COTS (veja abaixo) adquiridos, modificações especiais para teste ou ambientes operacionais, por exemplo, e controles de hardware com software embarcado.
 - ii. Conforme a ISO/IEC/IEEE 24765:2017, COTS é o acrônimo de *commercial-off-the-shelf*, aplicável tanto a software quanto a hardware. No caso de software, é um produto disponível para compra e uso sem a necessidade de conduzir atividades de desenvolvimento. Observe que esta definição também se aplica a software para os quais não se aplicam pagamentos ou considerações de licença comercial convencionais.
3. Produzir ou adaptar elementos de serviço de sistemas de software.
 - i. Elementos de serviço incluem um conjunto de serviços a serem fornecidos, por exemplo, serviços de comunicações de rede, treinamento, serviços de empacotamento e distribuição de software, serviços de personalização do software para atender necessidades específicas de clientes, monitoramento de segurança e de operação e assistência ao usuário.
4. Avaliar unidade de software e dados correlatos ou outra informação conforme critérios e estratégia de implementação.
 - i. Critérios para avaliação geralmente incluem critérios de teste, cobertura de testes de unidade, rastreabilidade de requisitos, consistência com requisitos de elementos de software ou projeto, consistência interna de requisitos da unidade e viabilidade para atividades de processo posteriores como integração, verificação, validação, operações e manutenção.

5. Empacotar e armazenar elementos de sistema de software.
 - i. Para software, em geral exige uma cópia mestre do software implementado (em meio eletrônico ou físico) e armazenado em um local controlado e disponibilizado para papéis autorizados para uso nos processos de integração e transição. Informações do produto e de configuração são capturadas pelos processos de gerenciamento de informação e gerenciamento de configuração quando o elemento é armazenado.
6. Registrar evidência objetiva de que o elemento do sistema de software satisfaz requisitos.
 - i. Evidências são fornecidas em conformidade com acordos, legislação e política organizacional. Evidência inclui modificações no elemento realizadas durante o processamento de mudanças ou não conformidades encontradas durante os processos de validação e verificação. A evidência objetiva é parte da baseline estabelecida pelo processo de gerenciamento de configuração e inclui os resultados de testes de unidade, análises, inspeções, demonstrações, revisões técnicas ou de produto ou outros exercícios de verificação.

Gerenciar resultados da implementação

1. Registrar os resultados da implementação e anomalias encontradas.
 - i. Inclui anomalias decorrentes da estratégia de implementação, da implementação de sistemas habilitadores, ou definições incorretas do sistema de software. Processos de garantia da qualidade e de controle e avaliação de projeto são usados para analisar os dados e identificar a causa raiz, permitir ações corretivas ou melhorias, e registrar lições aprendidas.
2. Manter a rastreabilidade dos elementos implementados do sistema de software.
 - i. Fontes de licenças de software e outros ativos da cadeia de suprimentos devem ser registrados. Os processos de gerenciamento de informação e de gerenciamento de configuração são usados para manter licenças e termos de suporte de manutenção para as aplicações e as infraestruturas exigidas (*host system*).
 - ii. Rastreabilidade bidirecional é mantida entre os elementos implementados e a arquitetura do sistema de software; projeto e requisitos relacionados, incluindo requisitos de interfaces e definições que são necessárias para a implementação; planos de verificação e validação, procedimentos e resultados.
3. Fornecer os artefatos e itens de informação incluídos em baselines.
 - i. O processo de gerenciamento de configuração é usado para estabelecer e manter itens de configuração e baselines. Para esse processo, os elementos de sistema de software como código fonte, pacotes de software e resultados de teste unidade são artefatos típicos que são depositados na baseline.

Referências

ISO/IEC/IEEE 12207:2008. [ISO/IEC/IEEE Standards for System and Software Engineering -- Software Life Cycle Processes](#). 2008.

ISO/IEC/IEEE 12207:2017. [ISO/IEC/IEEE Standards for System and Software Engineering -- Software Life Cycle Processes](#). 2017.

ISO/IEC/IEEE 24765:2017. [ISO/IEC/IEEE Systems and Software Engineering -- Vocabulary, 2017](#).