Métodos de Desenvolvimento de Software

Como Escolher o Processo a ser utilizado para o desenvolvimento de um produto?

Considerações... **ABORDAGENS ÁGEIS**

- Consideram o projeto (design) e a implementação como as atividades centrais no processo de software.
- Incorporam outras tarefas a essas atividades, como a elicitação dos requisitos e os testes.

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Considerações... ABORDAGENS DIRIGIDAS A PLANO

- Identificam etapas diferentes no processo de software, com resultados associados a cada uma delas.
- Esses dados são utilizados como base para o planejamento da atividade de processo seguinte.

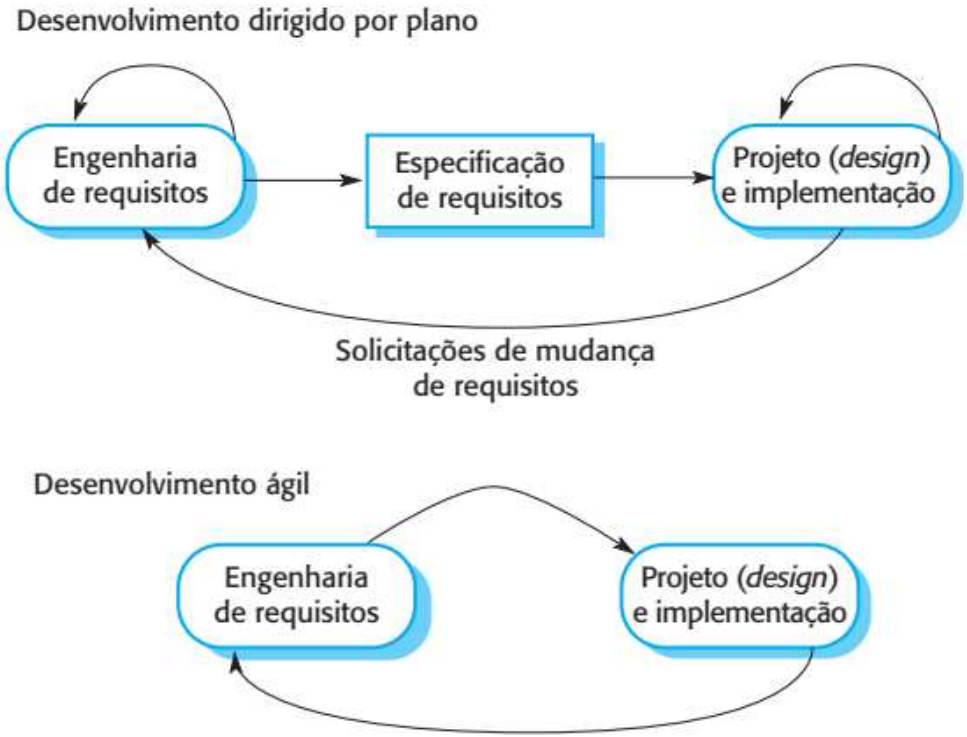
Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Considerações... (cont.)

- Em um processo de desenvolvimento de software **DIRIGIDO A PLANO**, a iteração ocorre dentro das atividades, com documentos formais sendo utilizados como meio de comunicação entre as etapas do processo.
  - Por exemplo: os requisitos evoluirão e, no final das contas, será produzida uma especificação deles, que servirá como entrada para o processo de projeto e implementação.
- Em uma abordagem **ÁGIL**, por outro lado, a iteração ocorre ao longo das atividades. Portanto, os requisitos e o projeto (design) são desenvolvidos juntos, e não separadamente.

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Considerações... (cont.)



Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Considerações... (cont.)

- Na prática, os processos **DIRIGIDOS A PLANO** são utilizados frequentemente com práticas de programação ágil, e os métodos ágeis podem incorporar algumas atividades planejadas, além da programação e dos testes.

- Em um processo dirigido a plano, é perfeitamente viável alocar requisitos e planejar a fase de projeto (design) e desenvolvimento como uma série de incrementos.

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Considerações... (cont.)

- Já um processo **ÁGIL** não é inevitavelmente focado no código e pode produzir alguma documentação de projeto (design).

- Nos métodos ágeis, os desenvolvedores podem decidir que uma iteração não produzirá código novo, mas sim modelos e documentação de sistema.

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Abordagens: Ágil e Dirigido a Plano

- Abordagens Dirigidas a Plano
  - Castaca, Spiral, RAD, Processo Unificado, etc.
- Abordagens Ágeis
  - Scrum, XP, FDD, DSDM, Crystal, AUP, etc.

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Abordagens: Ágil e Dirigido a Plano (cont.)

- Um requisito fundamental da escalabilidade dos métodos ágeis é integrá-los às abordagens dirigidas por plano.
- Pequenas empresas iniciantes podem trabalhar com planejamento informal e de curto prazo.
- Empresas grandes podem possuir maior necessidade de ter planos e orçamentos de mais longo prazo para investimento, pessoal e evolução comercial. Seu desenvolvimento de software deve apoiar esses objetivos, então o planejamento em longo prazo é essencial.

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Abordagens: Ágil e Dirigido a Plano (cont.)

Princípios ágeis e a prática organizacional.

Princípio	Prática
Envolvimento do cliente	Isso depende de ter um cliente disposto e capaz de investir tempo com o time de desenvolvimento e que possa representar todos os *stakeholders* do sistema. Muitas vezes, os representantes dos clientes têm outras demandas e não podem fazer parte do time de desenvolvimento em tempo integral. Nas situações em que existem *stakeholders* externos, como autoridades reguladoras, é difícil representar suas opiniões para o time ágil.
Acolher as mudanças	Pode ser extremamente difícil priorizar as mudanças, especialmente nos sistemas em que há muitos *stakeholders*. Geralmente, cada um deles atribui prioridades diferentes a mudanças diferentes.
Entrega incremental	As iterações rápidas e o planejamento de curto prazo do desenvolvimento nem sempre se encaixam nos ciclos de longo prazo do planejamento e marketing empresarial. Os gestores de marketing podem ter de conhecer as características do produto com vários meses de antecedência para preparar uma campanha eficaz.
Manter a simplicidade	Sob a pressão dos cronogramas de entrega, os membros do time podem não ter tempo para realizar simplificações desejáveis no sistema.
Pessoas, não processos	Membros do time podem não ter a personalidade adequada para o envolvimento intenso, o que é característico dos métodos ágeis, e, portanto, podem não interagir bem com os demais membros do time.

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Abordagens: Ágil e Dirigido a Plano (cont.)

- Para dar conta desses problemas, a maioria dos grandes projetos de desenvolvimento ágil de software combina práticas das abordagens dirigidas por plano e das ágeis.
- Alguns são basicamente ágeis e outros são praticamente dirigidos por plano, mas com algumas práticas ágeis.
- Para decidir sobre o equilíbrio entre uma abordagem e outra, é preciso responder a uma série de perguntas técnicas, humanas e organizacionais. Essas questões se relacionam com o sistema que está sendo desenvolvido, com o time de desenvolvimento e com as organizações que estão desenvolvendo e adquirindo esse sistema

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

ATENÇÃO !!!

A seguir, são apresentadas possibilidades de critérios para a seleção de ciclos de vida e processos de desenvolvimento de software.

NÃO É RECEITA DE BOLO !!!

Ou seja, não exime o engenheiro de software da responsabilidade de ter o conhecimento de cada modelo, assim como do contexto do produto que será desenvolvido, para que possa fazer a escolha adequada.

Abordagem Sommerville

Qual abordagem utilizar?
Como escolher?

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Como escolher a Abordagem?

- Sommerville (2018) propõe que para decidir sobre o equilíbrio entre uma abordagem e outra, é preciso responder a uma série de perguntas, basicamente, de três naturezas distintas (e complementares), sendo:
  - **Técnicas:** Se relacionam com o sistema que está sendo desenvolvido
  - **Humanas:** com o time de desenvolvimento
  - **Organizacionais:** com a organização que estão desenvolvendo e/ou adquirindo esse sistema

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Como escolher a Abordagem? (cont.)

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Como escolher a Abordagem? (cont.)

Questões **TÉCNICAS**

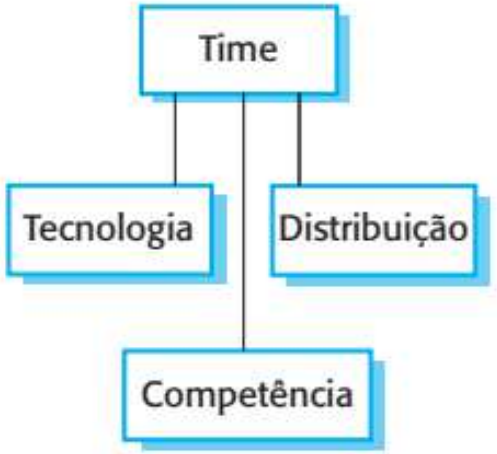
1) Qual é o tamanho do sistema que está sendo desenvolvido?
2) Que tipo de sistema está sendo desenvolvido?
3) Qual é a vida útil prevista para o sistema?
4) O sistema está sujeito a controle externo?

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Como escolher a Abordagem? (cont.)

Questões **HUMANAS**

1) Qual é o nível de competência dos projetistas e programadores do time de desenvolvimento?
2) Como está organizado o time de desenvolvimento?
3) Quais são as tecnologias disponíveis para apoiar o desenvolvimento do sistema?



Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Como escolher a Abordagem? (cont.)

Questões **ORGANIZACIONAIS**

1) É importante ter uma especificação e um projeto (design) bem detalhados antes de passar para a implementação — talvez por motivos contratuais?
2) É realista uma estratégia de entrega incremental, na qual o software é entregue aos clientes ou outros stakeholders e um rápido feedback é obtido?
3) Os representantes do cliente estarão disponíveis e dispostos a participar do time de desenvolvimento?
4) Existem questões culturais que possam afetar o desenvolvimento do sistema?

Sommerville, Ian Engenharia de software/ Ian Sommerville; tradução Luiz Cláudio Queiroz; revisão técnica Fábio Levy Siqueira. -- 10. ed. -- São Paulo: Pearson Education do Brasil, 2018. Título original: Software engineering ISBN 978-65-5011-048-2 1. Engenharia de software I. Siqueira,

Abordagem Gupta

Qual abordagem utilizar?
Como escolher?

Raja Gupta. Fundamentals of Software Engineering. Engineering Handbook. 2019.

Modelos de Ciclos de Vida

- **Modelo de ciclo de vida em CASCATA**
  - O modelo de ciclo de vida em cascata é o modelo de desenvolvimento de software mais direto como uma série de processos que devem ser executados apenas uma vez para o projeto.

Modelos de Ciclos de Vida

- **Modelo de ciclo de vida INCREMENTAL**
  - É semelhante à cascata em muitos aspectos, mas difere porque produz alguns resultados tangíveis para o cliente mais cedo.
  - Os processos iniciais de requisitos e viabilidade do sistema, requisitos de software e design geral são feitos em sequência, uma vez para o projeto geral.
  - Liberação antecipada de algumas partes do software.
  - Pode ser realizada por meio de várias equipes trabalhando em paralelo em diferentes incrementos.
  - Quando os requisitos são conhecidos e compreendidos, mas podem não ser estáveis.

Modelos de Ciclos de Vida

- **Modelo de ciclo de vida EVOLUTIVO**
  - Ao contrário dos modelos em cascata e incrementais, os processos de requisitos e viabilidade do sistema e requisitos de software **não** são feitos uma vez para todo o projeto, mas são revisados no início de cada ciclo de desenvolvimento evolutivo para incorporar os requisitos mais recentes.
  - É o modelo de ciclo de vida preferível quando os requisitos não são totalmente conhecidos, mas um subconjunto dos requisitos é conhecido, estável e compreendido.

Modelos de Ciclos de Vida

- **Modelo de ciclo de vida em ESPIRAL**
  - À primeira vista, o modelo espiral parece ser o desvio mais radical do modelo em cascata.
  - O principal benefício do modelo espiral está no controle dos riscos do projeto.
  - O uso do modelo espiral pode impedir a construção de software inadequado ou inutilizável, evitando retrabalho e aumentando a confiança entre os clientes e os desenvolvedores.
  - O gerente de projeto deve estar preparado para gerenciar do ponto de vista da prevenção de riscos. Os desenvolvedores, também, devem estar preparados para identificar, analisar e mitigar riscos – habilidade nem sempre identificada e cultivada.

Seleção de Modelo de Desenvolvimento de Software

- A seleção de um modelo de desenvolvimento de software adequado é baseada nas seguintes características/categorias:
  1) Requisitos
  2) Equipe de desenvolvimento
  3) Usuários
  4) Tipo de projeto e riscos associados

Características dos Requisitos

- Os requisitos são muito importantes para a seleção de um modelo adequado. Existem várias situações e problemas durante a captura e análise de requisitos.

Table 3.1 : *Selection of a model based on characteristics of requirements*

Requirements	*Waterfall*	*Prototype*	*Iterative Enhancement*	*Evolutionary development*	*Spiral*	*RAD*
Are requirements easily understandable and defined?	Yes	No	No	No	No	Yes
Do we change requirements quite often?	No	Yes	No	No	Yes	No
Can we define requirements early in the cycle?	Yes	No	Yes	Yes	No	Yes
Requirements are indicating a complex system to be built	No	Yes	Yes	Yes	Yes	No

Status da Equipe de Desenvolvimento

- Em termos de disponibilidade, eficácia, conhecimento, inteligência, trabalho em equipe etc., é muito importante para o sucesso do projeto.
- Se conhecermos os parâmetros e características da equipe acima mencionados, podemos escolher um modelo de ciclo de vida adequado para o projeto.

Table 3.2 : *Selection based on status of development team*

Development team	*Waterfall*	*Prototype*	*Iterative enhancement*	*Evolutionary development*	*Spiral*	*RAD*
Less experience on similar projects	No	Yes	No	No	Yes	No
Less domain knowledge (new to the technology)	Yes	No	Yes	Yes	Yes	No
Less experience on tools to be used	Yes	No	No	No	Yes	No
Availability of training, if required	No	No	Yes	Yes	No	Yes

Envolvimento do Usuário

- O envolvimento dos usuários aumenta a compreensão do projeto. Portanto, a participação do usuário, se disponível, desempenha um papel muito significativo na seleção de um modelo de ciclo de vida apropriado.

Table : 3.3 : *Selection based on user's participation*

Involvement of Users	*Waterfall*	*Prototype*	*Iterative enhancement*	*Evolutionary development*	*Spiral*	*RAD*
User involvement in all phases	No	Yes	No	No	No	Yes
Limited user participation	Yes	No	Yes	Yes	Yes	No
User have no previous experience of participation in similar projects	No	Yes	Yes	Yes	Yes	No
Users are experts of problem domain	No	Yes	Yes	No	No	Yes

Tipo de Projeto e Risco Associado

- Poucos modelos incorporam avaliação de risco. O tipo de projeto também é importante para a seleção de um modelo.

Table 3.4 : *Selection based on type of project with associated risk*

Project type and risk	*Waterfall*	*Prototype*	*Iterative enhancement*	*Evolutionary development*	*Spiral*	*RAD*
Project is the enhancement of the existing system	No	No	Yes	Yes	No	Yes
Funding is stable for the project	Yes	Yes	No	No	No	Yes
High reliability requirements	No	No	Yes	Yes	Yes	No
Tight project schedule	No	Yes	Yes	Yes	Yes	Yes
Use of reusable components	No	Yes	No	No	Yes	Yes
Are resource (time, money people etc.) scarce ?	No	Yes	No	No	Yes	No

Referências

- BOEHM, B. “A View of 20th and 21st Century Software Engineering” Proc. 28th Int. Conf. Softw. Eng. SE, ACM, pp. 12–29, 2006.
- CAPRETZ, F. L. (2014). Bringing the human factor to software engineering. Software, IEEE, 31(2), 104-104.
- CAPRETZ, L. F., AHMED, F., & DA SILVA, F. Q. B. (2017). Soft sides of software. arXiv preprint arXiv:1711.07876.
- MARSICANO, G., PEREIRA, D. V., DA SILVA, F. Q., & FRANÇA, C. (2017). Team maturity in software engineering teams. In Proceedings of the 11th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement (pp. 235-240). IEEE Press.
- Material de aula dos professores Hilmer Neri e Carla Rocha, FGA/UnB.

Referências (cont.)

- DINGSØYR, T. AND DYBÅ, T. (2012) “Team effectiveness in software development: Human and cooperative aspects in team effectiveness models and priorities for future studies”. In Proceedings of the 5th International Workshop on Co-operative and Human Aspects of Software Engineering (pp. 27-29). IEEE Press.
- LI, Z., AVGERIOU, P., & LIANG, P. (2015). A systematic mapping study on technical debt and its management. Journal of Systems and Software, 101, 193-220.
- SAWYER, S. (2004). Software development teams. Communications of the ACM, 47(12), 95-99.
- TAMBURRI, D. A., & DI NITTO, E. (2015). When Software Architecting Leads to Social Debt.
- TAMBURRI, D. A., KRUCHTEN, P., LAGO, P., & VAN VLIET, H. (2013). What is social debt in software engineering?. In Cooperative and Human Aspects of Software Engineering (CHASE), 2013 6th International Workshop on (pp. 93-96). IEEE.
- BROOKS, F. P. The Mythical Man-Month - Essays on Software Engineering. 20th Anniv ed. [S.l.]: Addison-Wesley, 1995.