

AULA 4 – MODELOS ITERATIVOS

OBJETIVO DA AULA

Compreender o modelo de desenvolvimento iterativo, apresentando as características, vantagens e desvantagens de modelos de desenvolvimento como Prototipação e Espiral.

APRESENTAÇÃO

Esta aula apresenta os modelos iterativos que tratam o desenvolvimento de software a partir de iterações, isto é, as atividades são repetidas para cada ciclo de desenvolvimento executado.

Dentre os modelos mais conhecidos desta fase estão o modelo chamado de prototipação e o modelo espiral.

No caso da prototipação, protótipos são construídos para a validação imediata junto ao usuário. Esta validação pode ajudar na definição da interface do software, na elicitação e validação de requisitos e também na construção de funcionalidades do software.

O modelo espiral define o desenvolvimento em quatro partes, definição de objetivos, avaliação e redução de riscos, desenvolvimento e validação, e planejamento. É um modelo que possui como destaque importante a validação e mitigação dos riscos do projeto.

1. OS MODELOS ITERATIVOS

Os modelos iterativos abordam a construção de software a partir da repetição das atividades do processo de software, conhecidas como iterações, até que o produto esteja disponível para ser disponibilizado para o uso do cliente.

É comum encontrarmos alguns modelos de desenvolvimento que juntam características dos modelos iterativos com os incrementais. Como o desenvolvimento e o gerenciamento de software é algo complexo, a ideia é dividir o trabalho em partes menores a partir de iterações, tendo como resultado de cada incremento.

2. PROTOTIPAÇÃO

O modelo conhecido como prototipação objetiva a construção de protótipos a cada ciclo iterativo, até que a repetição desse processo resulte na versão final do software.

A ideia é que esse processo agilize o desenvolvimento do produto, mas o foco está nas funcionalidades em si, e não no acabamento do produto, o que pode gerar versões incompletas e bastante retrabalho.

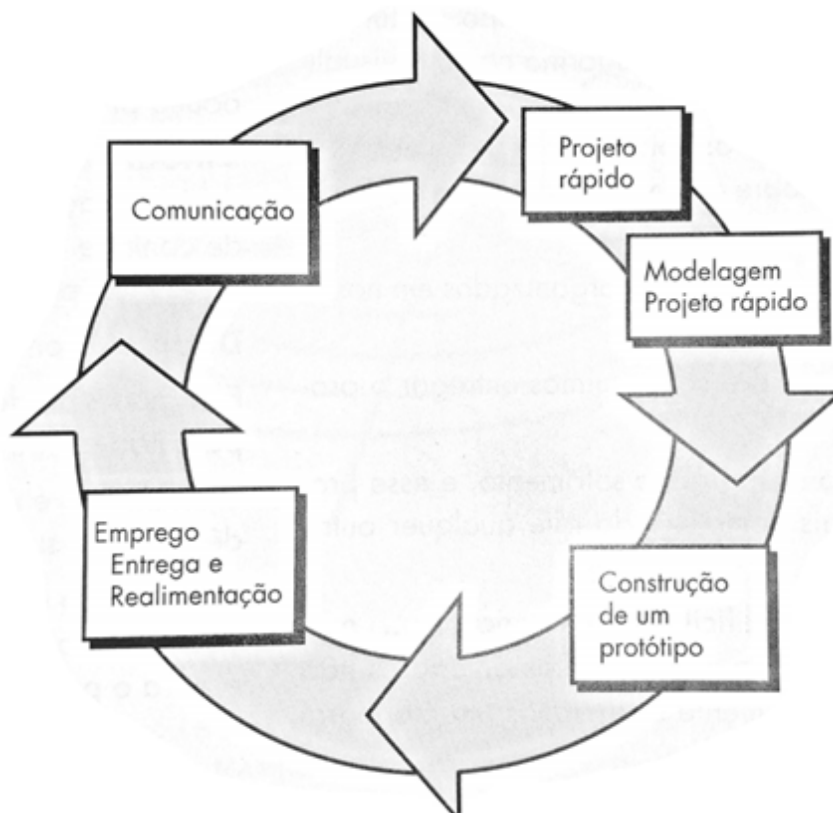
Livro Eletrônico

O modelo apresenta alguns problemas visíveis. Em primeiro lugar, o usuário tem a impressão de que aquele protótipo está sempre próximo à versão final do produto, e isso pode gerar alguns descontentamentos, pois a produção do protótipo normalmente não considera questões relacionadas à usabilidade e interface.

Em outro prisma, o usuário pode querer já usar o protótipo para resolver problemas críticos, não entendendo que ainda há muito trabalho a ser realizado até que o sistema esteja completamente disponível. De certa forma, deve-se evitar o uso desta abordagem em sistemas com maior criticidade.

Em outro ponto, a prototipação apresenta alguns benefícios observáveis. O primeiro diz respeito à constante interação com o usuário, que deve participar ativamente do processo, para aprovar os protótipos. O segundo diz respeito ao processo de validação imediata que este modelo requer, o que pode acelerar o processo de desenvolvimento como um todo.

FIGURA 1 | **O Paradigma da Prototipação**



Fonte: PRESSMAN (2019, p. 27).

Na figura 1 verificamos o ciclo iterativo da prototipação, em que o produto vai sendo construído a partir da execução cada ciclo completo.

Apesar de notavelmente apresentar alguns problemas, a prototipagem de sistemas possui algumas boas aplicações. Em casos em que o domínio é completamente desconhecido para o desenvolvedor, a sua aplicação ajuda a quebrar essas barreiras de entendimento do domínio.

Em engenharia de requisitos os protótipos também são bastante utilizados para ajudar na elicitac  o e valida  o dos requisitos do sistema que ser   desenvolvido. E neste ponto estamos falando do processo inicial de an  lise e especifica  o de sistemas.

Em um momento posterior, este tipo de abordagem de constru  o de software tamb  m pode ser utilizado para a defini  o de interfaces e usabilidade do sistema, essa aplica  o torna muito mais f  cil para o cliente aprovar ou solicitar ajustes na interface do software.

FIGURA 2 | **Processo de Desenvolvimento do Prot  tipo**



Fonte: SOMMERVILLE (2011, p. 30)

Na Figura 2 observamos o fluxo do ciclo de vida do prot  tipo executado at   sejam finalizadas todas as implementa  es levantadas na atividade de especifica  o de software.

Na fase inicial devem ser estabelecidos os objetivos do prot  tipo, que podem ser para validar a interface, validar requisitos ou para a demonstra  o da viabilidade do projeto de software.

A seguir deve-se estabelecer que requisitos ser  o inclu  dos o prot  tipo, no que chamamos de defini  o geral do prot  tipo. O desenvolvimento do prot  tipo vem logo a seguir, gerando ent  o um projeto execut  vel.

Por fim, h   a necessidade de avalia  o do prot  tipo. Nesta etapa um relat  rio deve ser constru  do visando apontar as falhas descobertas pelos usu  rios e os ajustes que devem ser feitos, como tempo de resposta e usabilidade, caso estas funcionalidades tenham sido definidas no plano de prototipa  o.

   importante destacar que os prot  tipos s  o   timas solu  es quando empregadas com outras t  cnicas. H   uma s  rie de softwares dispon  veis na web para a elabora  o de prot  tipos funcionais e de interface.

Prototipagem: o que    e 10 ferramentas para fazer o seu prot  tipo.
Dispon  vel em: <https://49educacao.com.br/mvp/prototipagem/>.
Acesso em: 02 nov. 2022.

LINK

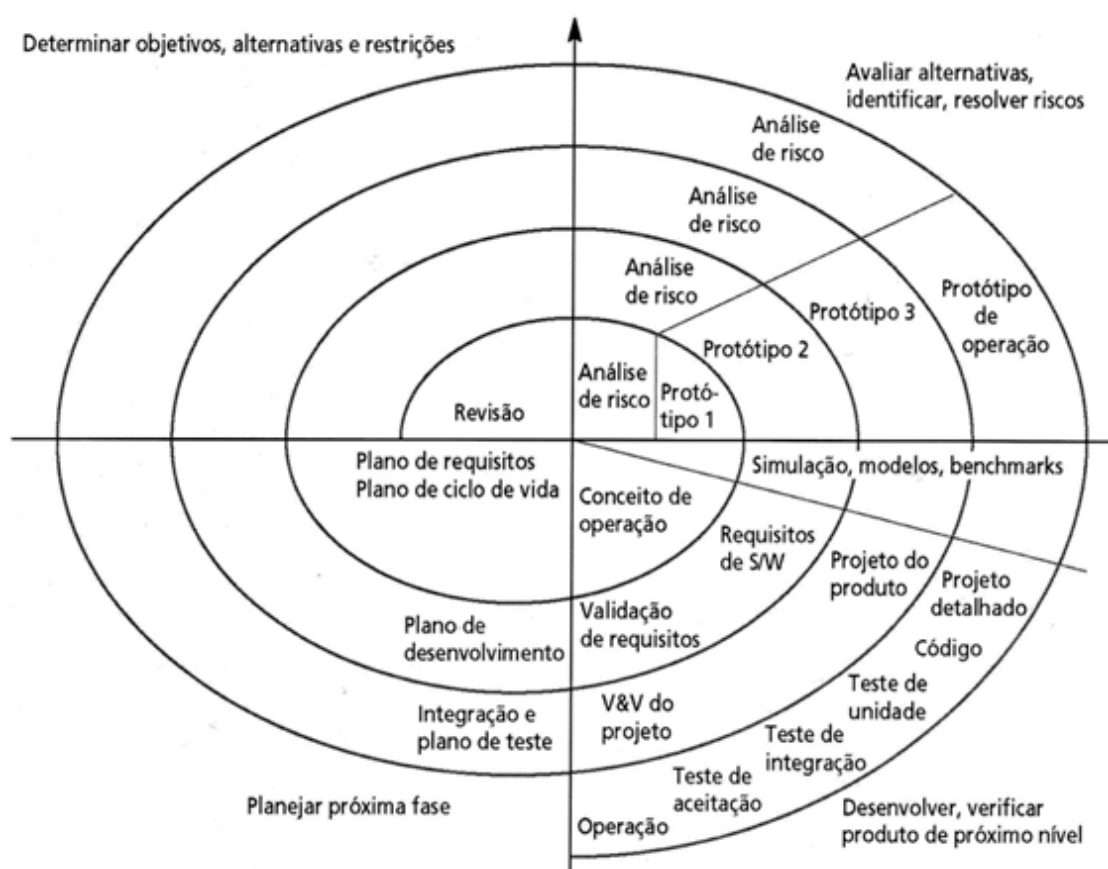


3. ESPIRAL

Alguns processos de software acabam agregando a união de paradigmas para a geração do seu próprio modelo. Este é o caso do modelo Espiral, concebido como uma tentativa da união do ciclo tradicional cascata com o modelo de protótipos, propondo o desenvolvimento em partes.

Este modelo possui quatro atividades: definição de objetivos, avaliação e redução de riscos, desenvolvimento e validação, e planejamento. É importante o entendimento de que estas fases possam ter nomes diferentes em algumas outras literaturas. A que abordaremos aqui se chama Modelo Espiral de Boehm.

FIGURA 3 | **Modelo Espiral de Boehm**



Fonte: SOMMERVILLE (2011, p. 33)

A Figura 3 especifica o modelo espiral considerando todas as atividades realizadas em cada uma das suas quatro fases.

Na fase de definição de objetivos, são definidos os objetivos, alternativas e são identificadas as restrições ao processo e produto. É necessária a construção de um plano de gerenciamento detalhado, assim como uma lista com os riscos do projeto que foram identificados.

Na fase de validação e redução de riscos, deve-se montar uma estratégia para mitigar cada risco encontrado na fase anterior. Em desenvolvimento e validação é definido um modelo de desenvolvimento do sistema.

A fase de Planejamento encerra o ciclo, e é o momento em que o projeto é revisado e o próximo giro no espiral é definido.

Apesar de ser um modelo realista, principalmente por conta da observação quanto aos riscos do projeto, o convencimento da sua adoção é difícil, pois o gerenciamento deste tipo de projeto não é trivial, justamente por conta das sucessivas evoluções. Deve-se considerar preocupar-se com uma análise de riscos muito bem elaborada.

CONSIDERAÇÕES FINAIS

Nesta aula vimos dois importantes modelos de elaboração de software chamados de Prototipação e Espiral. Esses modelos são baseados em iterações, ou seja, suas atividades são repetidas várias vezes até que o projeto seja finalizado.

O ciclo de vida prototipação trabalha com a implementação de protótipos que podem ser utilizados para diversos fins de validação junto ao cliente, como interfaces, requisitos e até na execução de funcionalidades.

Já o ciclo de vida em espiral requer dinamismo na implementação e experiência na definição, pois aborda o seu desenvolvimento com bases em iterações e incrementos. É um modelo importante por se preocupar em levantar e tratar os possíveis riscos de um projeto.

MATERIAIS COMPLEMENTARES

Vídeo: *Prototipação, saiba porque esta etapa é fundamental!* Disponível em: <https://www.youtube.com/watch?v=8YbAHNCv9-w>. Acesso em: 02 nov. 2022.

REFERÊNCIAS

PRESSMAN, R. G. *Engenharia de Software*. 9ª ed. Rio de Janeiro: McGraw-Hill, 2021.

SOMMERVILLE, I. *Engenharia de Software*. 10ª ed. São Paulo: Pearson Addison. Wesley, 2019.