

### Modelo Evolucionário: Espiral

Desenvolvido por Barry Boehm, o modelo espiral traz as ideias de controle do método cascata com as idéias de interatividade do método de prototipagem, como o próprio nome diz, em uma espiral, trazendo de início a comunicação com o cliente, a etapa de planejamento, etapa de modelagem, a construção do sistema e então cada entrega daquela espiral. Assim o sistema completo começa como algo simples e pequeno porém bem definido através dos protótipos e termina em algo grande e robusto, devido a ser um método bastante controlado o sistema acaba por ter riscos bem controlados. Porém o cliente pode ter mudanças de ideia sobre o início do projeto que podem causar problemas grandes.

### Desenvolvimento Ágil: XP

Desenvolvido por por Kent Beck, o modelo eXtreme Programming, traz consigo simplicidade, comunicação e feedback. Onde o sistema inteiro é baseado em, primeiro a etapa de planejamento com histórias de usuário, após isto é a etapa de projetamento utilizando em especial cartões CRC(classe-responsabilidade-colaborador), para então a etapa de codificação que funciona em duplas e em um loop interno de refabricação para alterações, por fim é feito a etapa de testes com testes de aceitação e testes de aplicação. As histórias de usuário são escolhidas com base em riscos e considerado seu valor, vendo o quanto isto impactará o sistema e assim de forma rápida e simples, onde cada história pode levar até 3 semanas, o sistema inteiro é construído. O problema é ser muito dependente do cliente e caso ele seja alguém ocupado pode atrasar o sistema.

	Modelo Evolucionário: Espiral	Desenvolvimento Ágil: XP
Flexibilidade e adaptabilidade	Apenas durante a fase de prototipação de cada espiral é possível realizar alterações no código, mas alterações em partes já concluídas requer reinício daquela espiral.	Devido a ser uma metodologia ágil, as alterações são muito fáceis, normalmente feitas durante a criação das histórias de usuário ou durante as etapas de fabricação que ocorrem frequentemente.
Abordagem para gerenciamento de riscos.	Para gerenciamento dos riscos é feito durante a etapa de planejamento, os mitigando com protótipos. Isto ocorre a cada nova espiral, requerendo consideração direta dos riscos para minimizá-los antes de se tornarem um problema.	Traz as histórias de usuário que podem causar algum risco como prioridade, e caso um risco aconteça, o método propõe utilizar uma "solução de ponta", basicamente um protótipo de uso único que será descartado após decidido.
Aplicabilidade em projetos de diferentes escalas e complexidades	Ótima metodologia para aplicar em um projeto muito grande, com um altos riscos e que precisa ser estável, porém leva mais tempo que metodologias ágeis para fazer quaisquer alterações, pois depende de	Ótimo para projetos pequenos ou projetos com um fluxo constante de alterações no projeto, porém demanda uma grande gestão para coordenar as diferentes equipes e as duplas de desenvolvedores. Devido a ser uma metodologia ágil, para

	uma nova revolução* na espiral.	pequenos projetos com equipe de desenvolvedores que não tem constantes mudanças, funciona bem. Porém a falta de documentação pode causar problemas para novatos nas equipes. Por fim, apesar de ter um tratamento de riscos, ele pode ser insuficiente para grandes projetos, pois caso um risco não previsto ocorra, é necessário fazer todo um protótipo que não terá uso.
Comunicação e envolvimento do cliente.	A Comunicação com o cliente ocorre durante a etapa de comunicação e de planejamento (onde é feito um protótipo), após isto o cliente fica de fora até a próxima revolução* da espiral	É feita constantemente, tanto que pode até se tornar um problema, pois o cliente pode trabalhar constantemente com a equipe de desenvolvimento criando novas histórias de usuários e dando feedback a cada refabricação; O problema que pode ser causado é devido ao cliente ser uma pessoa ocupada.

### C.3. Aplicação Prática em um Cenário de Projeto:

Usando o exemplo passado pelo professor, porém utilizando um sistema mais completo, acreditamos que a metodologia evolucionária espiral seja melhor. Pois é algo mais realista, onde consegue trazer feedbacks para o usuário e tomar suas mudanças, não desgasta os desenvolvedores e mantém uma boa documentação e especialmente, não toma tanto tempo do cliente. Devido a área de trabalho do cliente ser uma biblioteca é esperado que o sistema seja mantido por um longo tempo, então um sistema bem testado, otimizado, estabilizado e com uma fácil manutenção é essencial. O problema se mostra com o limite máximo de tempo, podendo até mesmo ser um risco importante, mas o sistema parece simples o suficiente para ser implementado na metodologia decidida.

Iniciaremos verificando com o cliente qual a sua principal preocupação, caso ele não tenha algo em mente, iniciamos com o protótipo do banco de dados que suporte uma grande quantidade de dados bem específicos e que deve ser mantido em um longo período de tempo. Afinal, nem todo dado será acessado a todo tempo, pode ocorrer que um dado específico nunca seja acessado. Então a escolha de algo como o MySQL se mostra interessante, pois de início é um projeto pequeno e facilmente escalável. Especialmente pois é de graça. Caso o cliente não tenha nada contra o uso do banco de dados escolhido ele passará para ser criado e testado.

Após o banco de dados, será verificado com o cliente se ele já tem alguma alteração em mente ou alguma prioridade a ser tomada, então, será feito o protótipo telas de preenchimento dos livros da biblioteca, onde o cliente pode cadastrar dados como nome, páginas, descrição, sinopse e classificação. Para então ser codificada, e testada junto com o banco de dados.

Após isto será feita a validação da tela de cadastro de livros, Após parte da equipe ficará responsável pelo protótipo de uma tela para que o bibliotecário consiga manter e remover livros em estoque. A outra parte da equipe ficará responsável por uma tela de login e cadastro de novos usuários. Com ambos os protótipos aprovados irá ser codificado e testado junto com o restante do sistema já pronto.

Novamente mostraremos o resultado atual e em seguida será feito um protótipo de menu para que o cliente da biblioteca possa acessar seu cadastro, modificá-lo caso algo esteja errado. Durante a criação desta etapa o bibliotecário comentou que gostaria de ter uma tela de históricos dos livros que cada cliente pegou da biblioteca. Devido a nossa metodologia usada, será necessário aguardar a próxima iteração da espiral. A tela de cadastro e manutenção de usuários foram feitas e testes de interação foram realizados.

Foram mostrado o resultado até então, para em seguida fazer o protótipo da tela de histórico, neste momento de prototipação o cliente fez outra alteração, solicitando uma tela de financeiro, onde o cliente da biblioteca possa fazer pagamentos das retiradas dos livros da seção paga. Devido a esta questão de alguns livros serem pagos, foi necessário prototipar a alteração na tela de cadastro de livros para indicar o valor de cada livro. Com todos os 3 protótipos aprovados, a equipe decidiu focar primeiro na tela de histórico devido ao tamanho curto, em seguida na alteração dos livros pois sem isto o sistema financeiro não funcionaria corretamente, por fim foi feita a tela de financeiro do cliente da biblioteca, mostrando os valores pagos, os devidos e o histórico de valores.

Foi mostrado o progresso até então e o cliente solicitou uma última tela e diz ele ser a mais importante, uma tela onde o cliente da biblioteca pode consultar os livros disponíveis, verificar seu valor, ver a sinopse e por fim agendar ou retirá-lo. O protótipo foi feito de acordo com a solicitação do bibliotecário e após aprovado a tela foi codificada. Para então ser realizado testes de integração desta tela e um teste completo do sistema.

O sistema acabou levando 5 meses para ser concluído e tinha previsão de 4, o risco foi causado devido ao cliente não ter especificado que alguns livros possuíam valores. Porém devido a gordura adicional no tempo o risco acabou não afetando o projeto.