

Atividade 2 – Estudo das metodologias de desenvolvimento de *software* para o projeto de biblioteca

Leia o enunciado da atividade e, com base nas informações do contexto, responda às seguintes questões:

1. Com relação à metodologia em cascata, quais as vantagens e desvantagens que essa metodologia apresenta? Considerando as características do sistema presentes no contexto do enunciado, há viabilidade de aplicar nele a metodologia em cascata?

O método conhecido como cascata, ou *Waterfall* não se encaixa nas especificações de versões intermediárias.

É uma metodologia tradicional que funciona bem em projetos pequenos e de pouca complexidade, mesmo fora da computação usamos esse método em algumas atividades. Como por exemplo ao nos deparar com um objeto de nossa casa que precisa de reparos e nós conseguimos repará-lo (Pendurar um espelho, trocar uma lâmpada por exemplo), no geral, nos deparamos com esse problema, analisamos, preparamos, implementamos e testamos e observamos, caso não fique seguro ou de nosso gosto, damos manutenção ou refazemos.

De qualquer forma, temos que ter em mente que essa metodologia não é tão dinâmica e não prevemos mudanças durante após a análise e definição, deixamos isso para outras metodologias.

Suas vantagens são: Simples aplicação e escopo menor e rápida aplicação.

Desvantagens: Não há aqui versões intermediárias do projeto, o objetivo é pensar, planejar e realizar o projeto, em projetos de longa duração e com alta taxa de mutabilidade essa metodologia não é indicada.

2. Descreva as vantagens e desvantagens da aplicação da metodologia ágil Kanban e comente a viabilidade para o projeto.

Como uma metodologia Ágil, o Kanban (カンバン) possui uma vantagem frente a metodologia de cascata, aqui nosso foco é separar em ações menores o todo que constitui esse projeto (nesse caso específico, já que essa metodologia vai muito além, idealizada pela Toyota em 1960, antes dos Post it!,

fora implementada nas linhas de produção), essas ações são divididas em listas, que normalmente são: Para fazer “*To Do*” | Executando “*Doing*” | Feito “*Done*”(Sempre me perguntei o motivo de ser em inglês e não em Japonês essas listas, já que veio do Toyotismo, tipo: すべきこと | やってる | 完成です), cada lista , normalmente, contém um número máximo de tarefas (pra não virar bagunça) e a cada reunião, alteração ou mesmo realização de tarefas(ações), as listas são atualizadas, dando um sentido de progressão e dinamismo ao projeto.

Suas vantagens são: A visualização de todo o projeto, a segmentação das ações e sua organização por prioridades. Pode ser uma ferramenta usada em outras metodologias (como *Scrum* pro exemplo, usa muito o Kanban)

Como uma forma de organizar, planejar e limitar cada tarefa e lista o Kanban é amplamente utilizado em várias fases de processos operacionais e administrativos, mas requer organização e disciplina para sua utilização, não é bem aproveitado em situações emergenciais, quando tudo é importante, nada é prioridade, nesse caso o Kanban não se aplica. (é só desespero e gritaria mesmo)

3. Considerando a metodologia *Scrum*:

a) Indique vantagens e desvantagens de utilizá-la no contexto apresentado.

A partir de uma *user stories*, o *Product backlog* é criado e atribuído ao *product owner*, que fará a *Sprint Plannig* e delimitará cada *sprint* e sua *deadline*. Enquanto o Time *Scrum* realiza *stand-up daily meeting* diariamente durante a *sprint* e cuida do desenvolvimento o *Scrum Master* trabalha para que todos entendam o processo *Scrum* e trabalhe com foco. Ao fim de cada *sprint* acontece a *Sprint review* em que o *Scrum Master*, *Product Owner* e o Time *Scrum* (único sem um termo chique) dão seus *feedbacks*. Por fim há ainda o *Sprint retrospective* onde o *Scrum Master* e o time *scrum* analisam a última *sprint* realizada e debater sobre potenciais pontos de melhoria para otimizar os processos da equipe.

Enfim, nesse contexto, percebemos que essa metodologia é mais “complexa”, não exatamente complexa é a palavra correta, mas ela é ideal para projetos de longa duração, que sofrem mudanças ao longo de seu desenvolvimento, possuem uma equipe maior e um produto mais complexo.

Sobre adaptabilidade, agilidade, objetividade, essa é uma das melhores metodologias para processos complexos (às vezes uma atividade é tão complexa que não sabemos nem por onde começar), o *Scrum* visa solucionar esses cenários, mas podemos nos perder em nossos papéis, às vezes o papel de um

desenvolvedor é impreciso e ele pode ficar sem saber o que fazer exatamente, outra situação comum são as reuniões diárias, nunca encontrei (em mais de 20 anos trabalhando) uma reunião que durasse menos de 30 min, o normal é degringolar em discussões que não fruem em nada.

Por fim, alguns prazos são impraticáveis, já ouvi colegas reclamando de *sprint* com dias de duração e complexidade de meses.

No geral o *Scrum*, caso bem aplicado, com a equipe na mesma página e um bom *Scrum Master* que resolva essas “picuinhas” é transformador e consegue entregar projetos muito bem feitos e com maior agilidade.

- b) Considerando a descrição da equipe, descreva quem poderia desempenhar cada um dos papéis previstos em uma rotina da metodologia Scrum.

Product Owner	Gerente da Área
Scrum Master	Alguém que conheça a metodologia, e saiba lidar com pessoas, pensei em colocar o gerente aqui, mas o PO é importante para separar as funcionalidades e essa pessoa tem o papel de liderança dentro do projeto. A pessoa Responsável pela biblioteca não cabe aqui pois não sei se ela conhece a metodologia e menos ainda saiba lidar com o Time de Desenvolvimento, enfim.
Desenvolvimento	Equipe de Dev. 5 Pessoas
Stakeholders	Responsável pela biblioteca, usuários da biblioteca, alunos...

- c) Considerando *sprints* de duas semanas, preencha a tabela a seguir com as cerimônias do *Scrum* que aconteceriam nos dias da *sprint*:

	Semana 1	Semana 2
Segunda-feira	<p>Sprint planning</p> <p>Criação das páginas e definição das funções de cada tipo de usuário (funcionários da biblioteca e usuários da biblioteca)</p>	<p>Daily Scrum</p> <p>Criação dos grupos de usuários (funcionários da biblioteca e usuários da biblioteca) e suas atribuições e permissões.</p> <p>Modificações ou criação dos design das páginas caso</p>

		necessário
Terça-feira	Daily Scrum Implementação do cadastramento das publicações	Daily Scrum Desenvolvimento dos relatórios de empréstimos.
Quarta-feira	Daily Scrum Entrega do primeiro protótipo e desenvolvimento do banco de dados (implementação do ISSN e/ou ISBN)	Daily Scrum Entrega do terceiro protótipo Retorno e ajustes caso hajam.
Quinta-feira	Daily Scrum Desenvolvimento dos registro de empréstimos. Modificações ou criação dos design das páginas	Daily Scrum Dia curinga (nunca se sabe se algo irá atrasar) Ajustes finais(caso hajam)
Sexta-feira	Daily Scrum Ajustes finais(caso haja) Sprint Review	Daily Scrum Ajustes finais(caso hajam) Sprint Review Sprint Retrospective Fim da <i>sprint</i> .

d) Partindo das prioridades do sistema, sugira três tarefas (*user stories*) para a primeira semana de desenvolvimento. Em cada tarefa, indique para quem servirá a funcionalidade e o motivo da tarefa.

Eu, como um:

Usuário

Quero:

Pesquisar um livro

De modo que:

Seja intuitivo, fácil e rápido

Eu, como um:

Funcionário

Quero:

Manutenção no sistema

De modo que:

Seja fácil, intuitivo e rápido.

Eu, como um:

Funcionário

Quero:

Ter acesso ao cadastramento de publicações, o registro de empréstimos e os relatórios de empréstimos

De modo que:

Sejam de fácil visualização, enxuto, rápido e responsivo.