**Ciclo de vida do software**

O Ciclo de Vida de Software é um conceito fundamental no desenvolvimento de software, que abrange todas as fases e atividades envolvidas desde a concepção de um sistema até a sua desativação. Contudo, existem diversas formas de desenvolver um programa e, consequentemente, diferentes modelos de ciclos de vida. Software Web ou Web Software é um programa que é utilizado pela Internet com um navegador web (Google Chrome, Safari, Firefox, etc), não sendo necessário efetuar um download (baixar) ou atualizá-lo. 3 figuras principais:

Líder técnico: coordena toda a equipe de programadores, auxilia nas soluções junto ao cliente e também na resolução de problemas. Ele seguido pelo próximo cargo da lista.

Analista sênior: responsável por apoiar o time de programadores/analistas

Gerente de projetos: responsável pela burocracia que afeta o projeto, como a parte administrativa, jurídica, etc.

1. ***Levantamento de requisitos***

Identificação dos requisitos do sistema, reunião do PO com os stakeholders (clientes), incluso controle de fornecedores, produção e venda, além dos demais requisitos que o cliente julgar essencial.

***2 Planejamento***

Reuniões de planejamento das Sprints para selecionar as User Stories (funcionalidades) a serem trabalhadas em cada Sprint, com base na priorização do Product Owner e na capacidade da equipe. Criação do Product Backlog que inclui todas as funcionalidades necessárias que o sistema deve ter no prazo final de acordo com os requisitos do sistema. Definição das metas e objetivos específicos para cada Sprint, considerando as necessidades de curto prazo da fazenda, como sazonalidade de produção e demanda do mercado. Nessa reunião, a equipe define o Sprint Backlog, que é uma lista de tarefas que serão trabalhadas durante a iteração (ou sprint) para que se atinja o objetivo principal (as tarefas do product backlog). O sprint backlog é o product backlog quebrados em sprints, cada sprint terá suas tarefas visando alcançar o objetivo final.

***3.Requisitos***

Durante a Sprint, a equipe analisa os requisitos detalhadamente. Os requisitos são divididos em tarefas menores, que são atribuídas aos membros da equipe. A comunicação contínua entre os membros da equipe é fundamental para garantir que todos entendam os requisitos e possam colaborar efetivamente.

***4. Desenvolvimento***

Realização das Sprints, com o desenvolvimento das funcionalidades selecionadas, seguindo o ciclo de trabalho do Scrum, que inclui a implementação, revisão, teste e integração contínuos. Realização de reuniões diárias (Daily Scrum) para sincronização da equipe, identificação de possíveis impedimentos e ajuste de planos conforme necessário. Implementação das funcionalidades definidas no Product Backlog. Priorização da colaboração e comunicação contínuas entre a equipe de desenvolvimento e os stakeholders para garantir que as necessidades do negócio sejam atendidas.

***5.Revisao e feedback***

Após o término de cada Sprint, realização de uma **Revisão da Sprint**, onde as funcionalidades desenvolvidas são apresentadas ao PO e posteriormente aos stakeholders para revisão e feedback. Incorporação do feedback recebido nas próximas iterações do ciclo de desenvolvimento, garantindo que o sistema evolua de acordo com as necessidades e expectativas dos usuários finais.

***6.Restrospectiva da Sprint***

Reflexão sobre o processo de desenvolvimento durante a Sprint. Identificação de pontos fortes e áreas de melhoria. Definição de ações para melhorar o próximo ciclo de desenvolvimento.

***7.Interaçao***

Repetição dos passos de planejamento, desenvolvimento, revisão e retrospectiva para cada Sprint subsequente. Priorização contínua do Product Backlog com base nas necessidades do negócio e no feedback do cliente.

***8.Iteraçao implantação***

Integração das funcionalidades desenvolvidas ao sistema existente. Testes finais para garantir a estabilidade e a qualidade do software. Lançamento da versão atualizada do sistema em produção

***9. Monitoramento e manutenção***

Acompanhamento do desempenho do sistema em produção. Correção de bugs e implementação de melhorias conforme necessário. Atualização do Product Backlog com novos requisitos e funcionalidades emergentes.

O ciclo visa assegurar a qualidade do produto final, atender às necessidades dos usuários e garantir a eficiência e eficácia do desenvolvimento. Permitir o controle de versões, realizar testes e manutenções adequadas, bem como garantir a documentação necessária.

Modelos de ciclo: cascata, incremental, espiral, evolutivo.

A partir das partes do desenvolvimento de software são comuns a todas elas. O sistema é concebido, são pensadas soluções automatizadas para melhorar a forma de realizar as atividades das fazendas. Logo, depois que os desenvolvedores escolher a metodologia do desenvolvimento das atividades de fornecedores, produção e venda de produtos ou prestações de serviços. Já com a interface feita, é feita a programação da aplicação com base no desenvolvimento, não podendo esquecer de fazer os testes de aplicação para evitar problemas na qualidade do alimento e a segurança alimentar, assim, evitar retrabalhos. Já com todas as partes definidas e concretizadas, entre a parte de operação e manutenção, em que o sistema este preparado para ser utilizado e em constante processo de manutenção e operação, podendo sempre implementar melhorias. Sendo o grande objetivo do ciclo de vida é para garantir não só a qualidade do sistema, mas também como o do produto que tal fazenda está fazendo.

<https://blog.brkambiental.com.br/fazendas-urbanas/#:~:text=Fazendas%20urbanas%20s%C3%A3o%20espa%C3%A7os%20concebidos,estufas%20agr%C3%ADcolas%20anexas%20%C3%A0s%20edifica%C3%A7%C3%B5es>.

<https://blog.cronapp.io/ciclo-de-vida-do-software/>

<https://uds.com.br/blog/ciclo-de-vida-do-software-web/>

b