

Gerente de TI

Pamella Gonçalves de Araujo

Como o próprio nome já diz, a principal função de um Gerente ou Gestor de TI é cuidar para que o setor de informática e tecnologia de uma empresa funcione como o esperado.

Entretanto com o crescente e atual avanço das tecnologias, o setor de TI vem deixando de ser apenas um setor operacional e se torna de extrema importância na tomada de decisões e no desenvolvimento dos objetivos das áreas de negócios.

As principais características que um profissional de TI precisa ter são:

- **Saber administrar todos os recursos dos setores de tecnologia**

Conhecimentos técnicos continuam sendo necessários e saber administrar é a principal tarefa. Organizar e planejar o uso e a aquisição de softwares de acordo com a necessidade da empresa. Saber criar e disponibilizar as vagas. Escolher os profissionais corretos. Além de definir e alinhar bem os times e recursos, para que os projetos ocorram como o planejado.

- **Ter boa capacidade de relação interpessoal e ser adaptável**

Participar de reuniões de negócios, entender as necessidades de outros colaboradores da empresa, fazer a comunicação entre os diversos setores e entender o que realmente o cliente precisa. Gestores que não possuem essas características enfrentam conflitos diariamente e acabam perdendo o foco de sua principal função.

- **Ser inovador**

Um profissional de TI deve e precisa sempre estar atualizado com o que acontece no mundo. Conhecer a posição de sua empresa no mercado, e como se posicionam os concorrentes. Deve estar ciente de todas as novas tecnologias, sempre apresentar novas ideias e melhorias que trazem ganho aos processos. O seu aprendizado tem de ser constante.

- **Ser um bom líder**

Apesar de todas as características e conhecimentos técnicos, um Gerente de TI deve saber como liderar. Definir vagas, contratar recursos e criar uma equipe (time) que confie nele. Além do mais, é a gestão dos colaboradores que irá determinar os bons resultados dos projetos.

Além da necessidade de estar em constante aperfeiçoamento, o Gerente de TI deve estar sempre atento às novas tecnologias que poderão impactar as rotinas operacionais, otimizar os processos e aumentar o grau de inovação dentro do ambiente corporativo.

Gestores adotam sistemas de planejamento para tornar o desenvolvimento dos projetos mais eficaz e ágil.

A seguir vamos conhecer alguns deles.

Lean

Como fazer mais com menos?

O conceito de Lean envolve a identificação e eliminação de desperdícios. Como fazer mais, melhor e com menos?

Baseado no sistema de produção automobilístico da Toyota, consiste basicamente em padronizar as funções. É uma filosofia operacional ou um sistema de negócios, que tem foco no que irá gerar valor. As ações são definidas e realizadas sem interrupção e apenas quando alguém as solicita. A principal base é realizá-las de forma mais eficaz possível, ou seja, evitando tudo o que não irá trazer valor.

O sistema Lean também se atenta ao desenvolvimento e envolvimento dos colaboradores. Com o padrão dos processos eles aprendem mais rapidamente e podem atuar em diferentes funções. Além disso é importante fazer com que se sintam valorizados e estimulados, evitando a evasão das empresas.

DDD

Domain Driven Design

O Domain Driven Design (Projeto Orientado a Domínio) é um conjunto de princípios que auxiliam os desenvolvedores a evitar problemas e aplicar práticas que melhoram o processo de desenvolvimento.

Para a metodologia DDD o mais importante é o software, não o código, ou as tecnologias usadas, mas a regra de negócio, o problema que ele se propõe a resolver.

Os padrões defendidos pelo DDD são divididos em 3 etapas:

1. Definir o contexto
2. Identificar o problema
3. Achar a solução

Para o modelo DDD funcionar corretamente ele deve estar baseado em:

- **Bom alinhamento do código com as regras de negócio.**

Os especialistas do negócio e o time de desenvolvimento devem falar a mesma língua. Termos cotidianos utilizados na linguagem de mercado devem ser identificados no códigos e deve-se evitar a ambiguidade de um mesmo termo.

- **Possibilidade de reutilização.**

Com o passar do tempo as regras se tornam cada vez mais familiares, tornando possível a refatoração e aperfeiçoamento do código. Fica fácil encontrar no código conceitos semelhantes que poderão ser reutilizados.

- **Mínimo acoplamento.**

Um sistema bem modelado pode ser integrado com diversos outros sistemas sem necessitar de grandes infra-estruturas para isso. Os módulos e entidades são distintos, com métodos próprios, possibilitando sua existência sem grandes dependências.

- **Independência de tecnologia.**

No modelo DDD o mais importante é o negócio e seu perfeito funcionamento independente do ambiente e das tecnologias usadas. A qualidade deverá ser a mesma. A preocupação do DDD é que as regras de negócio sejam contempladas no código.

Agile

Todo mundo é ágil

A abordagem de desenvolvimento Agile define um conjunto de práticas para desenvolver e ajudar outros no desenvolvimento de software.

Essa metodologia define 4 princípios chaves:

1. Os indivíduos são mais importantes que os processos.
2. O software em funcionamento vale mais que a documentação completa.
3. A participação e colaboração dos clientes interessam mais que contratos.
4. Adaptar-se a mudanças está acima de seguir os planos

A maior prioridade é satisfazer o cliente com entregas contínuas e de qualidade. Mudanças repentinas ao fim do desenvolvimento não causam graves danos ao escopo, permitindo aos clientes aproveitar vantagens competitivas, já que o processo se adapta a essas mudanças.

Especialistas de negócio e desenvolvedores trabalham junto, fazendo entregas em curtos períodos.

É importante formar um time de confiança, com boa comunicação, motivado e ciente de sua capacidade. Um time alinhado sempre busca a excelência e aumenta a agilidade.

Para garantir a qualidade dos métodos ágeis utiliza-se a prática de TDD (Test Driven Development - Desenvolvimento guiado por testes) que permite aplicar testes contínuos garantindo sempre a qualidade técnica do produto.

Scrum

A eficiência de um Sprint

Scrum é uma metodologia ágil para gestão e planejamento de software.

Os projetos são divididos em ciclos com duração semanais ou até mensais, denominados de **Sprints**. As funcionalidades que devem ser implementadas no projeto são mantidas numa lista chamada **backlog**.

Antes do início de cada sprint ocorre um **planning**, ou seja, uma reunião de planejamento que define quais tarefas do backlog devem ser executadas naquele sprint.

Diariamente o time faz reuniões rápidas, denominadas **daily**, nas quais os participantes apresentam o que resolveu entre um dia e outro, apontam impeditivos e assumem novas tarefas para o dia corrente.

Ao final de cada sprint, as funcionalidades implementadas são apresentadas em um **review**.

Personagens da metodologia Scrum:

- **Product Owner (PO) ou Dono do Produto**

Responsável por definir as funcionalidades do produto e priorizar as tarefas que devem entrar no sprint.

- **Scrum Master**

É o técnico do time. Garante que todas as práticas sejam seguidas, como daily, tempo de sprint, reviews, etc. Cuida para que não haja mais tarefas no sprint do que o time é capaz de cumprir. Além de cuidar do relacionamento do time com ambientes externos.

- **User Experience (UX) ou Designer**

Responsável pela usabilidade do produto. É a ponte entre o desenvolvimento e o usuário final.

- **Equipe de Desenvolvimento (Time)**

Quem faz o produto. É auto gerenciável, cada membro decide por qual tarefa ficará responsável e cabe ao próprio time cobrar a execução da mesma. O importante é que se tenha no time todos os perfis necessários para se fazer cumprir as tarefas.

- **Marketing do Produto**

É o responsável de levar o produto ao conhecimento das pessoas e encorajá-las a usá-lo durante o período de desenvolvimento. Isso ajudará na tomada de decisões da equipe de desenvolvimento.

- **Devops**

É sua responsabilidade codificar, documentar e manter a infraestrutura e o ambiente onde o sistema irá se manter vivo.

Kanban

Disciplina, Transparência e Priorização

É uma metodologia de desenvolvimento ágil, geralmente representado por um quadro onde cartões representam as tarefas a serem implementadas que irão seguir um fluxo definido com diferentes estágios.

O kanban implementa conceitos da Teoria das Restrições (TOC), onde cartões só podem ser colocados em circulação, quando este estiver disponível.

Diferentemente do Scrum, o Kanban, não tem um período de tempo definido. O modelo de produção é Just In Time, ou seja, as funcionalidades são implementadas assim que solicitadas.

O método Kanban é de simples visualização, com o diagrama de estágios, onde é possível localizar a posição de sua tarefa rapidamente. Por isso é importante os estágios serem bem definidos.

O fluxo mais comum segue a seguinte ordem:

TO DO => WIP (Work In Progress) => DONE

Os cartões (cards) podem ser classificados como storie, bug, test, etc. As classes ajudam na melhor organização do que está sendo produzido.

É importante priorizar tarefas ou definir metas, como por exemplo um tempo para cada estágio. Essas práticas garantem entregas de qualidade e clientes satisfeitos.

Deve-se sempre buscar a melhoria contínua e a perfeição, tornando-se útil a definição de métricas e simulação de riscos. É interessante saber a comparação do que é necessário fazer contra o que já foi produzido, como também criar planos preventivos diminuindo riscos.

Salários:

Porte da empresa	Trainee	Júnior	Pleno	Sênior	Master
Pequena	R\$ 3043.57	R\$ 3956.64	R\$ 5143.63	R\$ 6686.72	R\$ 8692.74
Média	R\$ 4565.35	R\$ 5934.96	R\$ 7715.45	R\$ 10030.09	R\$ 13039.12
Grande	R\$ 6848.04	R\$ 8902.45	R\$ 11573.18	R\$ 15045.13	R\$ 19558.67

População: 20733 salários Amostragem: 18051salário(s) 33contribuições.

Metodologia utilizada: salários pretendidos e contribuições salariais

Bibliografias:

<http://www.transportabrasil.com.br/2013/08/competencias-e-habilidades-de-um-gestor-de-ti/>

<https://www.impacta.com.br/blog/2015/05/06/quais-sao-as-responsabilidades-de-um-gerente-de-ti/>

<http://www.caminhandojunto.com.br/2012/02/sistema-lean-produzir-mais-com-menos.html>

<http://www.agileandart.com/2010/07/16/ddd-introducao-a-domain-driven-design/>

<http://www.manifestoagil.com.br/principios.html>

<http://www.concretesolutions.com.br/2014/03/26/os-seis-papeis-em-um-time-agil/>

<http://www.devmedia.com.br/kanban-4-passos-para-implementar-em-uma-equipe/30218>

<http://www.sine.com.br/media-salarial-para-gerente-de-ti>