[14:06, 13/6/2017] +55 11 97979-1584: Governança de TI é um conjunto de práticas, padrões e relacionamentos estruturados, assumidos por executivos, gestores, técnicos e usuários de TI de uma organização, com a finalidade de garantir controles efetivos, ampliar os processos de segurança, minimizar os riscos, ampliar o desempenho, otimizar a aplicação de recursos, reduzir os custos, suportar as melhores decisões e conseqüentemente alinhar TI aos negócios.

[14:06, 13/6/2017] +55 11 97979-1584: A governa de TI é considerada a “Gestão da Gestão” e tem como objetivo auxiliar o governante de TI a avaliar os rumos a serem tomados para alcançar os objetivos da organização, reunindo o desenvolvimento de um conjunto de habilidades e estratégias, para os profissionais de ti, garantindo controles efetivos, ampliação da segurança, mitigação de riscos otimizar desempenho e utilização de recursos.

O COBIT define:

• **Requisitos de negócio a serem atendidos**

• **Processos de TI que sustentam os negócios**

• **Recursos de TI que são aplicados na realização dos processos**

O COBIT não define COMO gerenciar TI mas sim O QUÊ deve ser feito para gerenciar TI.

**Ele define recomendações a serem seguidas pelo CIO na hora de definir suas práticas para gerenciar:**

• **Gerenciar pessoal e capacitação**

• **Compras e ativos sob a responsabilidade de TI**

• **Alinhar estratégias**

• **Segurança da informação**

• **Condução de projetos**

• **Manutenção e suporte técnico às operações**

**CONSEQUÊNCIAS** da falta de avaliação: **PARA EMPRESAS DE SOFTWARE**

• **Queda de credibilidade**

• **Redução da competitividade** pela perda de referências

• **Desmotivação e rotatividade da equipe** em função de cargas excessivas de trabalho em função de correções e adaptações em sistemas fornecidos pela empresa

• Recorrência de **esforços extraordinários** para ajustar o curso das coisas, com aumento de **custos**

• **Estresse no relacionamento com clientes** (litígios, renegociação e quebra de contratos, etc.)

**CONSEQUÊNCIAS** da falta de avaliação: **PARA O CLIENTE FINAL**

• **Percepção de falta de transparência** na comunicação por TI

• Sensação de **desamparo** em relação ao apoio que TI poderia fornecer ao negócio

• A **TI vira um centro de custo e não de negócios** e resultados

Esses **modelos difundem** práticas para gerenciar A **QUALIDADE TOTAL (TQM – TOTAL QUALITY MANAGEMENT).**

* **INSPEÇÃO**: Coleta de evidências e avaliação de resultados. Deve ser feito smp
* **CONTROLE DA QUALIDADE:** Monitoração e decisão sobre ação corretiva diante dos resultados
* **GARANTIA DA QUALIDADE**: Desenvolvimento de práticas que evitem problemas de qualidade. Não pode modificar a td momento. Mudanças devem ser planejadas
* **QUALIDADE TOTAL:** Desenvolvimento conjunto e permanente do recurso humano e processo de produção para atender as expectativas dos clientes que estão em constante mudança. Treinamento de pessoas p trabalhar com qualidade
* **GESTAO DO DESEMPENHO EMPRESARIAL:**

Governança, resultados financeiros, desenvolvimento de mercado, desenvolvimento da inteligência competitiva.

**VAMOS CONHECER MAIS SOBRE...**

**• ISO 9126 – Definição das dimensões avaliativas do software**

Esta norma estabelece as dimensões (chamadas CARACTERÍSTICAS e SUBCARACTERÍSTICAS) e atributos (chamados ATRIBUTOS e ATRIBUTOS BÁSICOS) de avaliação da qualidade de software.

**• SPICE ISO 15504 – Avaliação da capacidade do processo de software**

Define um programa de desenvolvimento da qualidade, estabelecendo nível de capacidade.Ficou conhecida como...**SPICE**

Documento organizado em áreas de trabalho ou de processos de TI as quais são exploradas em termos das suas práticas (tarefas a serem cumpridas, métodos e técnicas), ferramentas de trabalho empregáveis, capacitação para o trabalho (mão-de-obra e fornecimentos) e objetos resultantes do trabalho.

**Niveis de capacidades:**

5 – Processo melhorado continuamente

4 – Processo gerenciado quantitativamente

3 – Processo definido, praticado e acompanhado

2 – Processos praticados informalmente

1 – Sem práticas padronizadas de processos

**CMMi**

**• CMMi – Modelo de desenvolvimento e avaliação de Capacidade e Maturidade no desenvolvimento de software**

A iniciativa de documentar as melhores práticas em gestão da qualidade de software por parte do SEI surgiu no início dos anos 90 com o então chamado CMM (Capability Maturity Model). Capacidade = conseguir fz , maturidade= ter feitos mts vezes

Com o tempo, o **CMM ganhou diversos modelos para atender projetos, engenharia de software, engenharia de sistemas, aquisição de software, desenvolvimento da força de trabalho**, posteriormente integrados no CMMi (Capability Maturity Model Integration) em junho de 2000. Atualmente o CMMi está na versão 3, editada em 2010.

**Proposta do CMMi:**

• CMMi funciona como guia de boas práticas.

• **Confirmar o alcance da maturidade na gestão da qualidade da produção de software** –

**Níveis de Maturidade definido no guia:**

|  |  |
| --- | --- |
| **5** | **OTIMIZADO -** Foco na melhoria contínua de processos |
| **4** | **QUANTITATIVAMENTE GERENCIADO -** Processos são estatisticamente medidos e controlados |
| **3** | **DEFINIDO -** Processos estão documentados; Existe organização; Atividades são proativas no desenvolvimento de software |
| **2** | **GERENCIAD -** Orientação por projetos; Atividades são reativas no desenvolvimento de software |
| **1** | **INICIAL - P**rocessos não estão claramente definidos; Resultados são imprevisíveis; Atividades são reativas e pouco ou nada controladas |

**MPS.br – Modelo brasileiro para a produção de software com qualidade**

Criado no final de 2003, tem foco em micro, pequenas e médias empresas, apresentando um custo menor para evoluir nos estágios da qualidade.

Desvantagem = so valido no Brasil.

Esse Sistema Softex garante um eficiente auxílio nas áreas operacional, de financiamento e de capacitação das empresas associadas por meio de uma ampla e sólida articulação de parceiros da iniciativa privada, governo e academia.

---

**O desenvolvimento de processos empresariais busca:**

• Eficácia: se é feito aquilo que a empresa pede que seja feito;

• Eficiência: se as atividades são feitas com otimização de recursos;

• Efetividade: se estão sendo atendidas as expectativas dos clientes do trabalho.

-------------------------------

NA **LINHA CLÁSSICA**, existe uma **preocupação grande com a formalização** de atividades e as etapas do projeto são bem definidas e pouco flexíveis. (documentação, padrões) Em geral são **empregadas em situações de projetos com** escopo fechado ou situações de implantação de software que envolvem praticamente parametrizações (configurações) de instalação, **com baixa probabilidade de mudança em escopo, prazo e custo** inicialmente planejados. Descrição bem detalhada. Adotar em prijetos q não tem previsão de revisão em fases já encerradas.

**Método clássico + difundido: RUP RATIONAL UNIFIED PROCESS**

Métodos **Clássicos** em geral **seguem** os modelos de Ciclo de Vida **Cascata ou Incremental**.

**O método Clássico mais difundido é o RUP.**

**- cada fase eh subdividida em iterações,**

**- um conjunto de artefatos(release) é gerado a cada iteração**

**- um milestone(marco) é gerado em cda fase**

NA **LINHA ÁGIL**, o **foco está na geração de um produto útil**. Os **controles** sobre o projeto são **simplificados** para que o foco não se perca. Aceita-se mudanças ao longo do projeto, as quais envolverão **renegociação e replanejamento rápido**.

Métodos **Ágeis** geralmente **seguem** o Ciclo de Vida **Espiral ou Prototipação.**

Os projetos conduzidos sob uma **metodologia ágil** focam na **geração de valor para o cliente**, ou seja, procuram dedicar **maior atenção, tempo e capital na produção daquilo que o cliente vai usufruir**, sem dispender recursos demasiados em controles acessórios.

Foco na entrega do produto final, documentação necessária,

**“Manifesto para o Desenvolvimento Ágil de Software”**

· **Indivíduos e interações** acima de processos e ferramentas. // · **Software funcionado** acima de documentação abrangente.

· **Colaboração** com o cliente acima de negociação de contratos. // · **Responder a mudanças** acima de seguir um plano.

Atualmente, os métodos ágeis ganham grande espaço no mercado por:

• Atenderem rapidamente as expectativas dos clientes; // • Focarem equipes de desenvolvimento naquilo que gostam de fazer

• Aumentarem as taxas de sucesso quanto a qualidade do projeto em termos de processo e produto

**OPEN UP:**

O OPENUP **defende os princípios fundamentais de agilização** de projetos de software que incluem:

• entregas por pacote em tempos definidos. Incrementos são as atividades feitas p serem entregues

• Desenvolvimento do produto de forma incremental e evolutiva

• Facilidade e simplicidade na solução

• Adaptação ao longo do projeto

• reunião diária e curta falando sobre acontecimentos também do dia anterior

Ele **mantém a organização da equipe** definida por papéis e responsabilidades distintos definidos no RUP:

• possui mts funções, o que não vai a favor das metodologias ágeis

• burocracia

**RAD MODEL**

MÉTODOS ÁGEIS – DSDM/RAD

O **RAD (Rapid Application Development) e o DSDM (Dynamic System Development Method)** são métodos **baseados no ciclo de vida Incremental** e partem do princípio de que o software deve ser desenvolvimento de forma a serem criados e **reaproveitados seus vários componentes** e módulos.

Eles também pregam a adoção de **ferramentas de geração e validação de código** e de avaliação qualitativa de componentes do software.

**O ciclo de vida** no RAD é dividido em **Modelagem de Negócio**, Modelagem de **Dados**, Modelagem **Funcional**, **Geração da Aplicação, Teste e Entrega**.

O problema desses métodos está na **necessidade de se ter uma definição precisa dos componentes de software (especificação) antes de iniciar** um desenvolvimento de um sistema integrado

**XP PROGRAMMING**

**Primeiro método ágil a dominar o mercado. Menos tarefas adm e a equipe faz tds as etapas dos desenv.O desenvolvedor desenvolve e testa.**

**Desenvolver o proj em pares. Os 2 fazem o msm projeto, definição em dupla de desenvolvedores p ter feedback instantaneo**

**Valores do XP**

**Comunicação**:// **Simplicidade**v//**Feedback** // **Coragem**

**MÉTODO ÁGIL SCRUM**

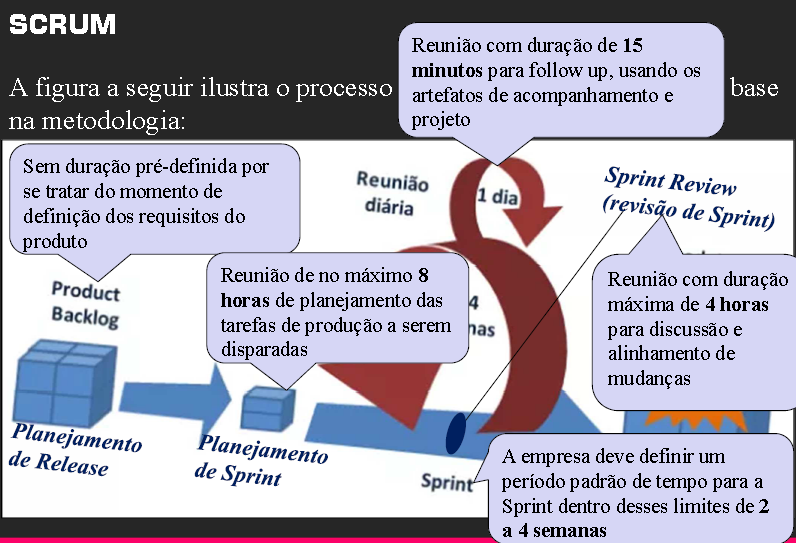
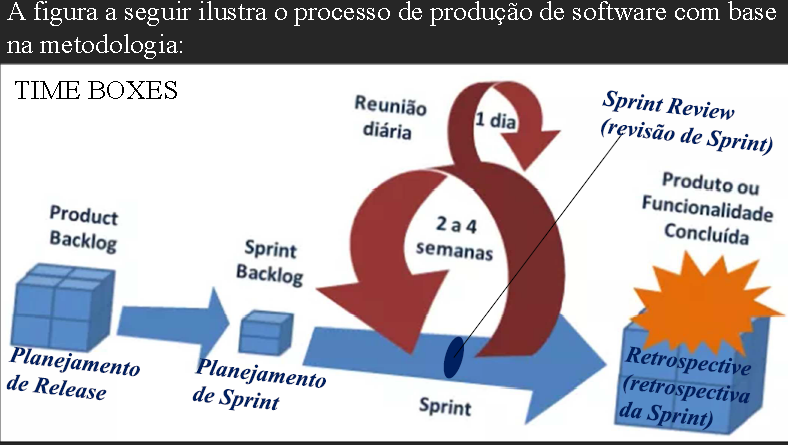
O SCRUM traz um conceito forte de trabalho em TIME para concretizar resultados rápidos, tendo clara definição de papéis e

responsabilidades!

Ele é organizado em um processo definido em TIME BOXES, que são caixas de tempo pré-fixado, mantendo assim um ritmo de produção,

tendo o compromisso de gerar frequentemente uma entrega útil para o cliente.

Ele inclui ainda, um conjunto de ARTEFATOS de gestão, que nada mais são que documentações gerenciais padronizadas a serem produzidas ao longo do ciclo de processo produtivo



**PLANEJAMENTO DE RELEASE/ PRODUCT BACKLOG :** Não tem tempo determinado.gera o prouct backlog, lista de pendencias a serem construídas p o soft**;.** Sem duração pré-definida por se tratar do momento de definição dos requisitos do produto.// Gera o plano de requisitos a serem entregues para que se tenha um produto/solução completo. OUTPUT (SAIDA): Produt backlog

**Backlog de produto é uma planilha ou quadro de pendências, semelhante a que é elaborada na engenharia de requisitos, quando da análise e desenho de sistema, da engenharia de software.**

**PLANEJ DE SPRINT:** Reunião de no máximo **8 horas** de planejamento das tarefas de produção a serem disparadas // Gera um plano de tarefas com atribuição de cartões de trabalho para membro do time, com definição de prioridade e peso do trabalho. OUTPUT: Sprint backlog. No plan de Sprint eu desmembro em tarefas o que eu defini no “planej de release;” ex, se no plan de release eu defini que teria q “cadastrar clienete”, neste item de plan da Sprint, eu defino as tarefas dentro de “cadastrar cliente”, como por ex: prototipar, modelar BD, modelar UC, codificar,. Etc tc e atribuo tarefas as pessoas.

Cada Sprint terá um **conjunto de Cartões de Trabalho que correspondem as tarefas** a serem realizadas para **produzir os itens do Backlog de Produto previstos** para desenvolvimento e entrega na Sprint!

**REUNIAO DIARIA**: Reunião com duração de **15 minutos** para follow up, usando os artefatos de acompanhamento e projeto // Observa e ajusta os artefatos de registro de evolução do projeto de forma ilustrada, simples e direta

**SPRINT:** A empresa deve definir um período padrão de tempo para a Sprint dentro desses limites de 2 a 4 sem // Executa as tarefas planejadas mudando o status de realização dos cartões de trabalho.

**SPRINT REVIEW:** Reunião com duração máxima de **4 horas** para discussão e alinhamento de mudanças // Ajusta plano em curso,redistribuindo, adicionando ou eliminando tarefas para que se acomode mudanças, mantendo o TimeBox

**RESTROSPECTIVE**: Reunião de até 3 horas para debate de lições aprendidas que serão úteis para as SPRINTS que se seguirão. // Guarda e versiona o pacote de produto liberado (fontes, documentos).

**Dentro da SPRINT acontece:**

Busca da ***Eliminação ou Minimização de Perda***, que é tudo aquilo em um processo ***que não agrega valor***.

Por exemplo, o tempo que um documento está na mesa de alguém aguardando alguma ação ou a própria construção de um documento que não será usado em tomadas de decisão).

**SCRUM**

A equipe de um projeto SCRUM tem a seguinte distribuição de papéis e responsabilidades:

**Product Owner:** Define a necessidade de negócio

Responsável por garantir o ROI (Retomo de Investimento)

Responsável por conhecer as necessidades do(s) cliente(s)

**SCRUM Master:** Capacita e avalia a aplicação da metodologia SCRUM

Ensina a doutrina do scrum

Responsável por remover os impedimentos do time

Responsável por garantir o Uso de Scrum

Protege o time de interferências externas

**Time de Desenvolvimento** : Produz a solução e gerencia o projeto

Definir metas das iterações

Autogerenciamento

Produzir produto com qualidade e valor para o cliente

**SCRUM Artefatos**

A aplicação do SCRUM demanda os seguintes materiais para documentação de apoio ao gerenciamento:

* ***Product backlog***
* ***Sprint backlog - (task cards – cartões de trabalho)***
* ***Quadro de tarefas - (KANBAN Board e Sprint BurnDown Graph***
* ***Histórias de usuários***