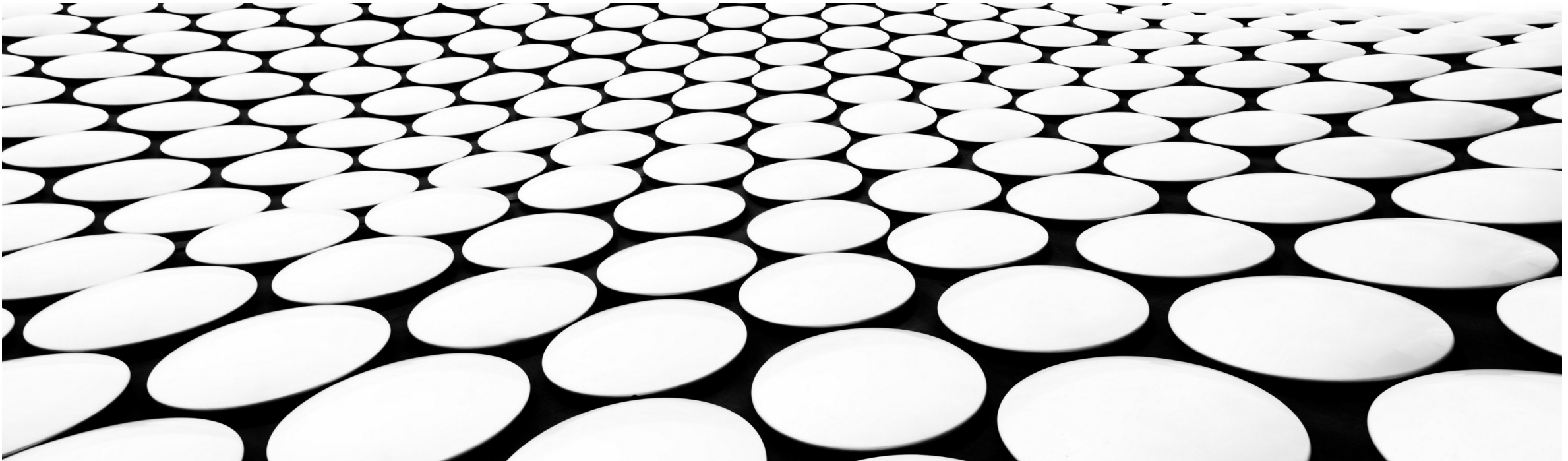

GERENCIA DE PROJETO DE SOFTWARE

CAROLINA MARIA FRANCISCO COTA





REFERÊNCIAS

- PRESSMAN, Roger S., MAXIM, Bruce R.. Engenharia de Software: uma abordagem profissional. 8ed. Porto Alegre: AMGH 2016.



QUESTÕES A SEREM RESPONDIDAS

- Como as pessoas, os processos e os problemas devem ser gerenciados durante um projeto de software?
- Como as métricas de software podem ser usadas para gerenciar um projeto de software e o processo de software?
- Como uma equipe de software pode gerar estimativas confiáveis de trabalho, custo e duração do projeto?
- Quais técnicas podem ser usadas para avaliar os riscos que podem impactar o sucesso do projeto?
- Como um gerente de projeto de software seleciona o conjunto de tarefas de engenharia de software?
- Como é criado um cronograma de projeto?
- Por que a manutenção e o suporte são tão importantes para os gerentes de engenharia de software e para os profissionais?

OS 4 P'S

- **Pessoas:** elemento mais importante para o sucesso do projeto.
- **Produto:** o software a ser construído.
- **Processo:** o conjunto de atividades e tarefas de engenharia de software que precisam ser feitas para se alcançar os objetivos.
- **Projeto:** todo trabalho realizado para tornar o produto uma realidade.

AS PESSOAS ENVOLVIDAS (STAKEHOLDERS)

- **Gerentes seniores** (*product owners*): que definem os problemas do negócio que, com frequência, têm influência significativa no projeto.
- **Gerentes de projeto - técnicos** (*scrum masters* ou líderes de equipe): que devem planejar, motivar, organizar e coordenar os programadores que executam o trabalho de software.
- **Profissionais**: que têm as habilidades técnicas necessárias para desenvolver a engenharia de um produto ou aplicação.
- **Clientes**: que especificam os requisitos para o software a serem submetidos ao processo de engenharia, e outros envolvidos que têm interesses periféricos no produto final.
- **Usuários**: que interagem com o software depois que ele é disponibilizado para uso operacional.

TIMES DE SOFTWARE



LÍDER DE EQUIPE

- **Seja o modelo.** Os líderes devem praticar o seu discurso.
- **Inspire e crie uma visão compartilhada.** Os líderes reconhecem que não é possível liderar sem seguidores. É importante motivar os membros de equipe.
- **Questione o processo.** Os líderes devem tomar a iniciativa de procurar maneiras inovadoras de melhorar o seu próprio trabalho e o das suas equipes.
- **Capacite os outros a agir.** Crie confiança e facilite relacionamentos para promover as habilidades de colaboração da equipe.
- **Incentive o espírito.** Comemore as conquistas dos indivíduos. Comemore as metas e vitórias compartilhadas, dentro e fora da equipe, para fortalecer o espírito de comunidade (de equipe).

EQUIPE DE SOFTWARE

- Os seguintes fatores devem ser considerados ao selecionar uma estrutura de equipe de projeto de software:
 - dificuldade do problema a ser resolvido.
 - “tamanho” do programa (ou programas) resultante, em linhas de código ou pontos de função.
 - tempo durante o qual a equipe vai permanecer reunida (vida da equipe).
 - até que ponto o problema pode ser modularizado.
 - qualidade e confiabilidade exigidas do sistema a ser construído.
 - flexibilidade da data de entrega.
 - grau de sociabilidade (comunicação) exigida para o projeto.

EQUIPE DE SOFTWARE

- Independente da organização da equipe devemos procurar montar uma equipe **coesa**.
“Uma equipe consistente é um grupo de pessoas tão fortemente unidas que o todo é maior do que a soma das partes...”
- Uma vez que uma equipe começa a ser consistente, a probabilidade de sucesso aumenta muito.
 - A equipe pode se tornar imbatível, um rolo compressor de sucesso...
 - Não é preciso gerenciá-la do modo tradicional e, com certeza, ela não precisará ser motivada.
 - Ela adquire ímpeto.

TOXICIDADE DE EQUIPE

- Uma atmosfera de trabalho frenética.
- Alto grau de frustração que causa atrito entre os membros da equipe.
- Um processo de software “fragmentado ou coordenado de forma deficiente”.
- Uma definição nebulosa das funções dentro da equipe de software.
- Contínua e repetida exposição a falhas.

TIMES ÁGEIS

■ Relembrando:

- a filosofia ágil enfatiza a satisfação do cliente e a entrega prévia incremental de software.
- pequenas equipes de projetos altamente motivadas e “auto organizada”.
- métodos informais, mínimos artefatos de engenharia de software e total simplicidade de desenvolvimento.

PRODUTO - ESCOPO DO SOFTWARE

- **Contexto.** Como o software a ser desenvolvido se ajusta a um sistema, produto ou contexto de negócio maior e quais são as restrições impostas como resultado do contexto?
- **Objetivos da informação.** Quais objetos de dados visíveis ao cliente são produzidos como saída do software? Quais objetos de dados são necessários como entrada?
- **Função e desempenho.** Qual função que o software executa para transformar os dados de entrada em dados de saída? Há quaisquer características especiais de desempenho a serem tratadas?

DECOMPOSIÇÃO DO PROBLEMA

- Às vezes chamado de particionamento ou elaboração de problemas.
- Uma vez que o escopo é definido...
 - É decomposto em funções constituintes.
 - É decomposto em objetos de dados visíveis ao usuário.
 - É decomposto em um conjunto de classes de problemas.
- O processo de decomposição continua até que todas as funções ou classes de problema tenham sido definidas.

PROCESSO

- Uma vez que uma estrutura de processo tenha sido estabelecida:
 - Considere as características do projeto.
 - Determine o grau de rigor necessário.
 - Defina um conjunto de tarefas para cada atividade de engenharia de software
 - Conjunto de tarefas =
 - Tarefas de engenharia de software.
 - Produtos de trabalho.
 - Pontos de garantia de qualidade.
 - *Milestones* (marcos do projeto).

COMBINANDO O PRODUTO COM O PROCESSO

- **O PROJETO** começa com a junção do produto com o processo.

| ATIVIDADES METODOLÓGICAS COMUNS DO PROCESSO | Comunicação | | | Planejamento | | | Modelagem | | | Construção | | | Entrega | | |
|--|-------------|--|--|--------------|--|--|-----------|--|--|------------|--|--|---------|--|--|
| Tarefas de engenharia de software | | | | | | | | | | | | | | | |
| Funções de produto | | | | | | | | | | | | | | | |
| Sincronizar telefone com dispositivo | | | | | | | | | | | | | | | |
| Visualizar dados no telefone | | | | | | | | | | | | | | | |
| Definir metas pessoais | | | | | | | | | | | | | | | |
| Armazenar dados na nuvem | | | | | | | | | | | | | | | |
| Permitir que o usuário modifique a interface do telefone | | | | | | | | | | | | | | | |
| Integrar com mídias sociais | | | | | | | | | | | | | | | |
| Definir metas com amigos | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |
| | | | | | | | | | | | | | | | |

Figura 24.1
Integração do problema com o processo.

PROJETO – INDICADORES DE PERIGO

- O pessoal de software não entende as necessidades de seus clientes, o que leva a um projeto com escopo mal definido.
- As alterações são mal gerenciadas.
- A tecnologia escolhida ou as necessidades do negócio mudam e perde-se o apoio da direção.
- Os prazos finais não são realistas.
- Os usuários podem se opor ao novo sistema (são resistentes).
- A equipe de projeto simplesmente não tem as habilidades necessárias.
- Os gerentes evitam as melhores práticas de lições aprendidas (necessário aprender com os erros).

PROJETO – CARACTERÍSTICAS DE SUCESSO

- Requisitos claros e fáceis de entender, aceitos por todos os envolvidos.
- Participação ativa e contínua dos usuários durante todo o processo de desenvolvimento.
- Um gerente de projeto com as habilidades de liderança necessárias, capaz de compartilhar a visão de projeto com a equipe.
- Um plano de projeto e um cronograma desenvolvidos com a participação dos envolvidos para atingir os objetivos dos usuários.
- Membros de equipe habilidosos e engajados.

PROJETO – CARACTERÍSTICAS DE SUCESSO

- Membros da equipe de desenvolvimento com personalidades compatíveis e que gostam de trabalhar em um ambiente colaborativo.
- Estimativas orçamentárias e de cronograma realistas que são monitoradas e mantidas.
- Necessidades dos clientes que são entendidas e satisfeitas.
- Membros de equipe com altos níveis de satisfação no trabalho.
- Um artefato operacional que reflete a qualidade e o escopo desejados.

O PRINCÍPIO DO 5W2H

- (**Why**) Por que o sistema está sendo desenvolvido?
- (**What**) O que será feito?
- (**When**) Quando será feito?
- (**Who**) Quem será o responsável por uma função?
- (**Where**) Onde se posicionam organizacionalmente?
- (**How**) Como será realizado o trabalho técnica e gerencialmente?
- (**How much**) Quanto de cada recurso (pessoas, softwares, ferramentas, banco de dados, servidores) será necessário?