

The slide features abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex arrangement of overlapping translucent green triangles and polygons creates a layered, architectural effect. The main text is centered on a light gray background.

Processos de software

Prof. Nelson Bellincanta Filho



Como o cliente explicou...



Como o lider de projeto entendeu...



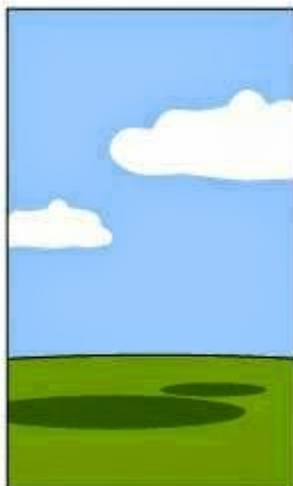
Como o analista projetou...



Como o programador construiu...



Como o Consultor de Negócios descreveu...



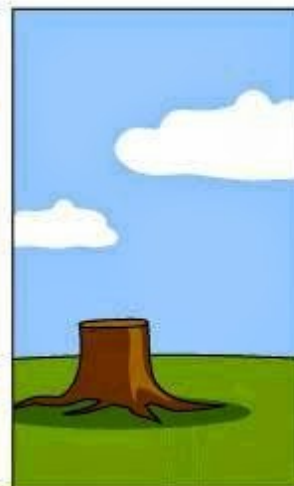
Como o projeto foi documentado...



Que funcionalidades foram instaladas...



Como o cliente foi cobrado...



Como foi mantido...



O que o cliente realmente queria...

Processo de software

- ▶ Conjunto de etapas, as quais são compostas por uma série de tarefas;
- ▶ Pode envolver o desenvolvimento de software a partir do zero, em uma determinada linguagem de programação;
- ▶ A ampliação e a modificação de sistemas já em utilização pelos usuários.

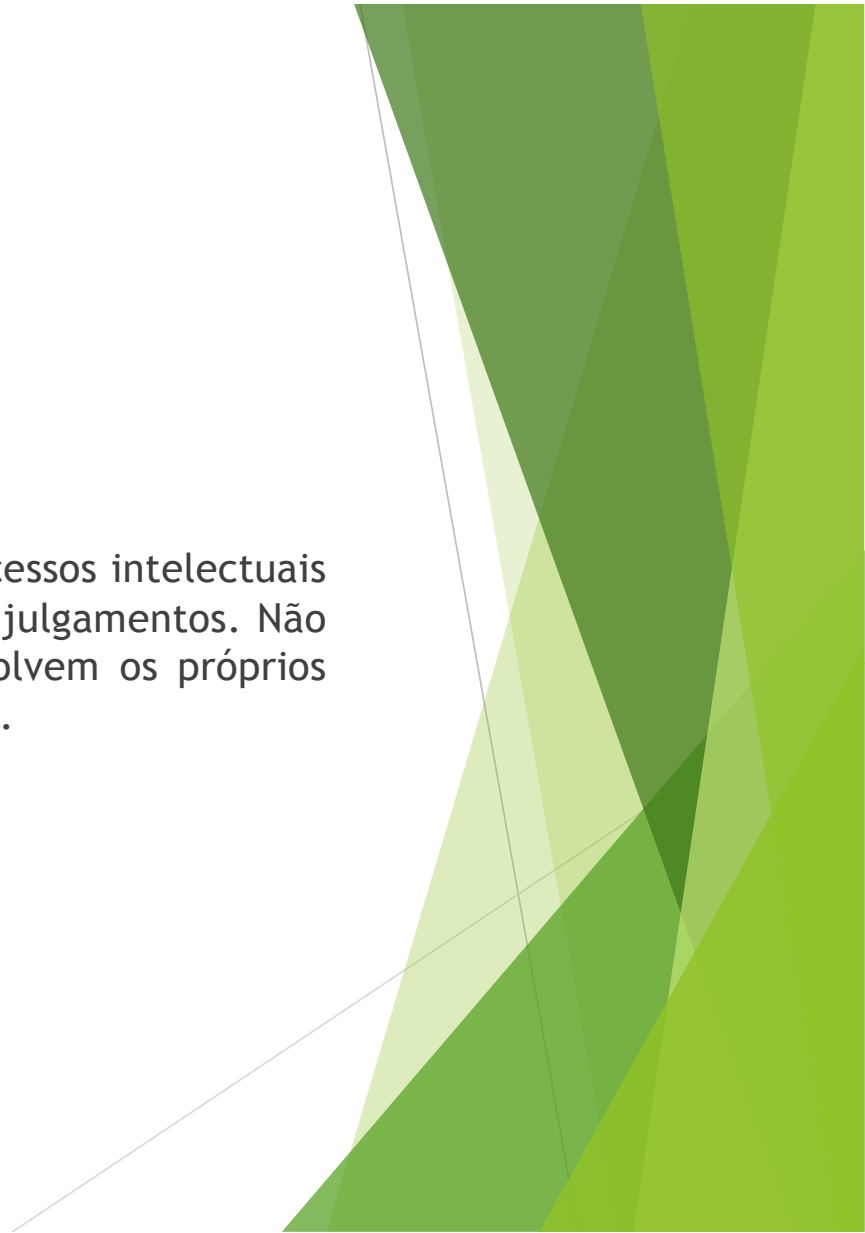
Processo de software

- ▶ Processos de software devem incluir:
 - ▶ Especificação de software;
 - ▶ Projeto e implementação de software;
 - ▶ Validação de software;
 - ▶ Evolução de software.



Processo de software

"Os processos de software são complexos e, como todos os processos intelectuais e criativos, dependem de pessoas para tomar decisões e fazer julgamentos. Não existe um processo ideal, a maioria das organizações desenvolvem os próprios processos de desenvolvimento de software." Sommerville (2018).



Modelos de Processo de Software



Modelos de Processo de Software

- ▶ Representação simplificada de um processo de software;
- ▶ Incluem as atividades que fazem parte do processo de software, os artefatos de software que devem ser produzidos em cada uma das atividades (documentos) e os papéis das pessoas envolvidas na engenharia de software.

Modelos de Processo de Software

- ▶ Waterfall (ou em cascata);
- ▶ Ágeis (ou incrementais ou iterativos).



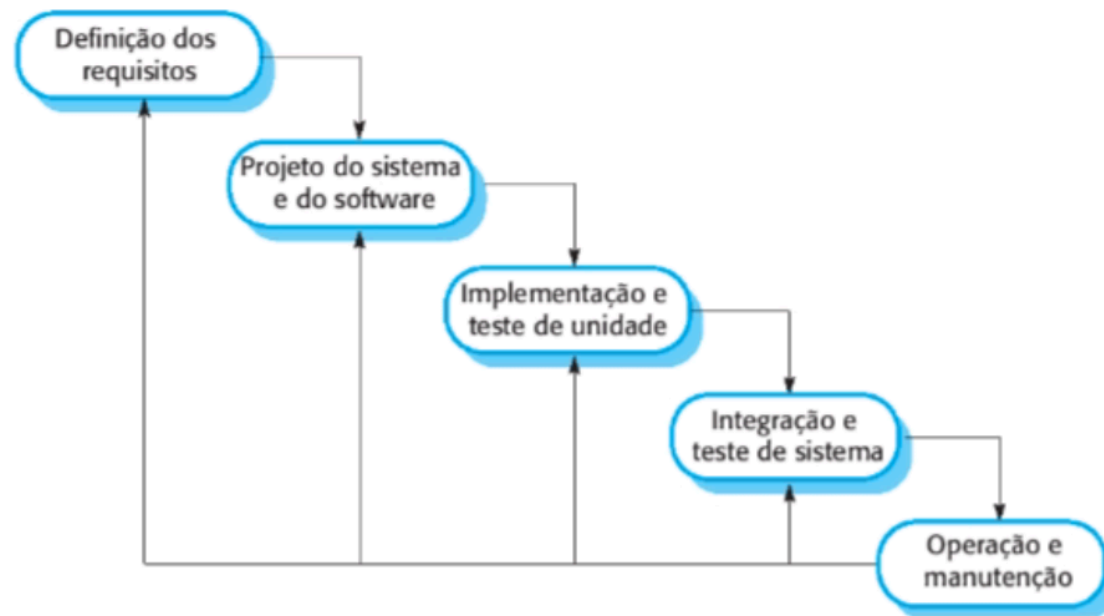
The slide features a light gray background with abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex arrangement of overlapping translucent green triangles in various shades of green is positioned. The text is centered in a green, sans-serif font.

Modelo Waterfall (ou modelo em cascata)

Modelo Waterfall

- ▶ Paradigma mais antigo da engenharia de software;
- ▶ Devido à cascata de uma fase para outra, esse modelo é conhecido como modelo em cascata ou ciclo de vida do software;
- ▶ Abordagem sequencial e sistemática para o desenvolvimento de software;
- ▶ Os estágios do modelo waterfall refletem diretamente as atividades fundamentais do desenvolvimento de software;
- ▶ Um estágio só pode ser iniciado depois que o anterior tenha sido concluído.

Modelo Waterfall



O modelo waterfall.
Fonte: SOMMERVILLE, Ian., 2018

Modelo Waterfall

► Problemas

- As empresas dificilmente seguem o fluxo sequencial proposto pelo modelo;
- O cliente não consegue definir claramente todas as suas necessidades;
- Versão operacional do sistema somente estará disponível no final do projeto.

Modelo Waterfall

Segundo Sommerville (2018) e Pressman (2011), o modelo waterfall deve ser utilizado somente quando os requisitos são fixos e com pouca probabilidade de alteração durante o desenvolvimento do sistema.



Modelo Waterfall

- ▶ Adequado somente para alguns tipos de sistema:
 - ▶ Sistemas embarcados, nos quais o software deve interagir com sistemas de hardware;
 - ▶ Sistemas críticos, nos quais há necessidade de ampla análise da segurança (safety) e segurança da informação (security) da especificação e do projeto do software;
 - ▶ Grandes sistemas de software, que fazem parte de sistemas de engenharia mais amplos, desenvolvidos por várias empresas parceiras.

Modelo Waterfall

- ▶ Não é recomendado para situações em que a comunicação informal do time é possível e nas quais os requisitos de software mudam rapidamente;
- ▶ Para esses sistemas, o desenvolvimento iterativo e os métodos ágeis são melhores.

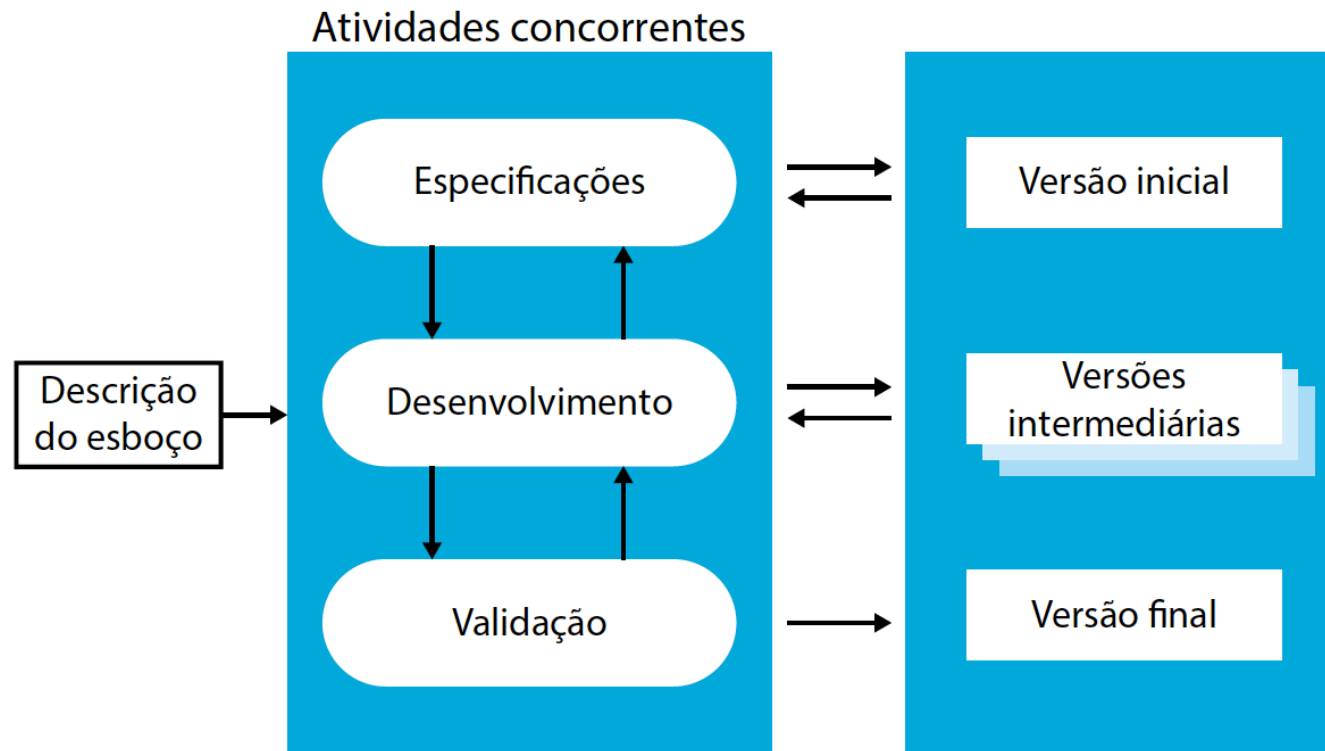
The background features abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex arrangement of overlapping translucent green triangles and polygons creates a layered, architectural effect. A thin, light gray line extends from the bottom left towards the right, passing through the green shapes.

Desenvolvimento incremental

Desenvolvimento incremental

- ▶ Desenvolver uma implementação inicial, obter feedback dos usuários ou terceiros e fazer o software evoluir através de várias versões, até alcançar o sistema necessário;
- ▶ As atividades de especificação, desenvolvimento e validação são intercaladas, em vez de separadas, com feedback rápido ao longo de todas elas.

Desenvolvimento incremental



Modelo de Desenvolvimento Incremental.
Fonte: Sommerville (2018).

Desenvolvimento incremental

- ▶ Abordagem mais comum para o desenvolvimento de softwares de aplicações e produtos;
- ▶ Essa abordagem pode ser dirigida por plano ou ágil; na maioria das vezes, uma mistura de ambas;
 - ▶ abordagem dirigida por plano, os incrementos do sistema são identificados antecipadamente;
 - ▶ abordagem ágil, os incrementos iniciais são identificados, mas o desenvolvimento dos incrementos finais depende do progresso e das prioridades do cliente.

Desenvolvimento incremental

- ▶ O desenvolvimento incremental de software é parte fundamental dos métodos ágeis;
- ▶ Melhor do que uma abordagem em cascata para os sistemas cujos requisitos estão propensos a mudar durante o processo de desenvolvimento;
- ▶ Desenvolver um software de modo incremental, é mais barato e fácil fazer alterações nele durante o processo de desenvolvimento.

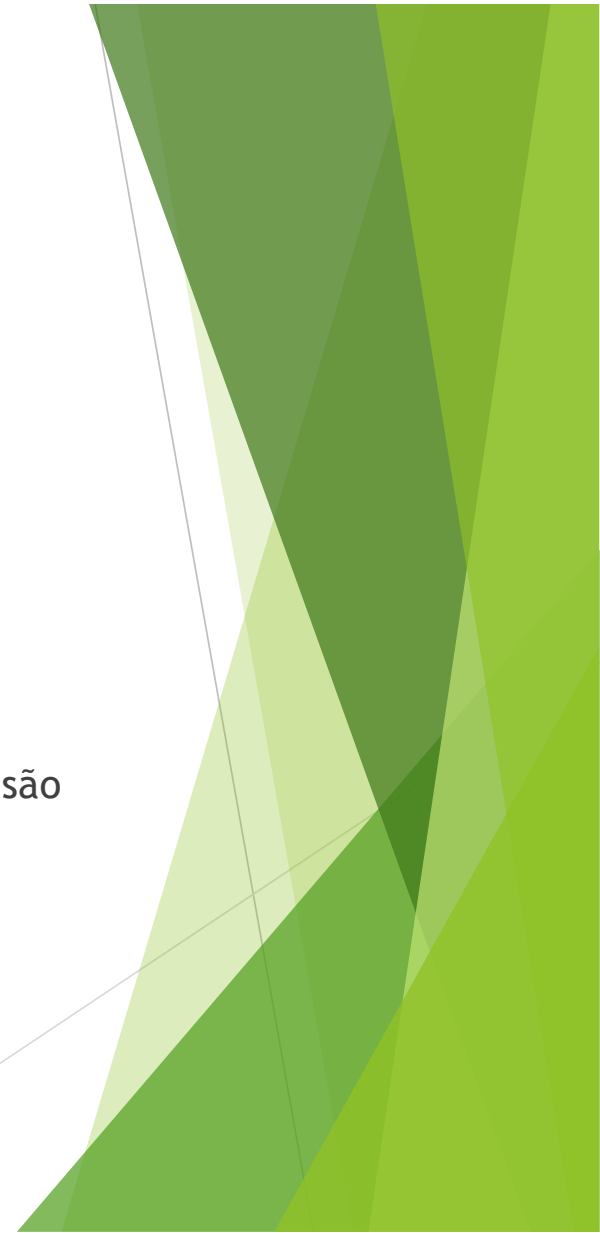
Desenvolvimento incremental

- ▶ Cada incremento ou versão do sistema incorpora parte da funcionalidade necessária para o cliente;
- ▶ Geralmente, os incrementos iniciais incluem a funcionalidade mais importante ou a mais urgente;



Desenvolvimento incremental - Vantagens

- ▶ O custo de implementação das mudanças nos requisitos é reduzido;
- ▶ Mais fácil obter feedback do cliente sobre o trabalho de desenvolvimento;
- ▶ A entrega e a implantação antecipadas de um software útil para o cliente são possíveis, mesmo se toda a funcionalidade não tiver sido incluída.



Problemas do desenvolvimento incremental

- ▶ Do ponto de vista da gestão, a abordagem incremental tem dois problemas:
 1. O processo não é visível;
 2. A estrutura do sistema tende a se degradar à medida que novos incrementos são adicionados;
 3. Podem ser exigidas ferramentas e técnicas especiais: elas possibilitam rápido desenvolvimento, mas podem ser incompatíveis com outras ferramentas ou técnicas. Além disso, relativamente, poucas pessoas podem ter a habilitação necessária para utilizá-las.

Problemas do desenvolvimento incremental

- ▶ Os problemas do desenvolvimento incremental se tornam particularmente críticos nos sistemas grandes, complexos e de vida longa, nos quais diferentes times desenvolvem partes distintas do sistema;
- ▶ Adotar o desenvolvimento incremental não significa ter de entregar cada incremento para o cliente.

Quando utilizar o modelo de desenvolvimento incremental

- ▶ Para sistemas pequenos (com menos de 100 mil linhas de código) ou para sistemas de porte médio (com até 500 mil linhas de código), com tempo de vida razoavelmente curto, a abordagem incremental de desenvolvimento talvez seja a melhor opção.
- ▶ Contudo, para sistemas de grande porte, os quais possuem longo tempo de vida e várias equipes desenvolvem diferentes partes do sistema, os problemas de se utilizar o desenvolvimento incremental tornam-se particularmente graves.
- ▶ Para esses sistemas, é recomendado um processo misto, o qual incorpore as melhores características do modelo de desenvolvimento em cascata e do incremental, ou, ainda, algum outro modelo disponível na literatura.

The background features abstract green geometric shapes. On the left, a solid green triangle points downwards. On the right, a complex arrangement of overlapping translucent green triangles in various shades of green and yellow-green creates a layered effect. A thin, light gray line extends from the bottom right towards the center of the page.

Integração e configuração

Integração e configuração

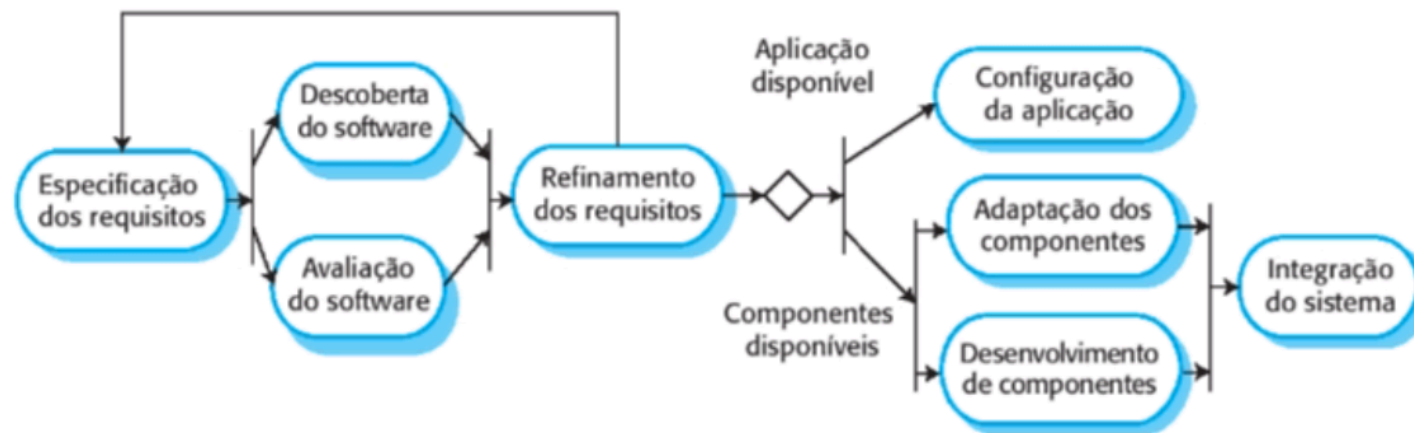
- ▶ Na maioria dos projetos há algum reuso de software;
- ▶ Acontece informalmente quando as pessoas que trabalham no projeto conhecem ou procuram algum código similar ao necessário;
- ▶ Ocorre independentemente do processo de desenvolvimento utilizado;



Integração e configuração

- ▶ Três tipos de componentes de software são reusados frequentemente:
 1. Sistemas de aplicação stand alone configurados para utilização em um ambiente particular;
 2. Coleções de objetos desenvolvidos como um componente ou como um pacote a ser integrado a um framework de componentes;
 3. Web services desenvolvidos de acordo com os padrões de serviço e que estão disponíveis para uso remoto na internet.

Integração e configuração



Integração e configuração

- ▶ A engenharia de software baseada no reuso, articulada em torno da configuração e da integração, tem a vantagem óbvia de reduzir a quantidade de software a ser desenvolvido, diminuindo custos e riscos;
- ▶ Permite uma entrega mais rápida do software;
- ▶ Entretanto, concessões quanto aos requisitos são inevitáveis, o que pode resultar em um sistema que não satisfaz as necessidades reais dos usuários.

Referências

- ▶ PRESSMAN, R. S.; MAXIM, B. R. **Engenharia de software: uma abordagem profissional**. 8. ed. Porto Alegre: AMGH, 2016.
- ▶ SOMMERVILLE, Ian. **Engenharia de Software**. 10ª Edição. 2018.
- ▶ VALENTE, Marco Tulio. **Engenharia de software moderna. Princípios e Práticas para Desenvolvimento de Software com Produtividade**. 2020.