

Universidade Federal de São Carlos - UFSCar

João Vitor Azevedo Marciano 743554

Exercícios Aula 20/04 - Modelos de Processos

Ciência da Computação - Introdução à Sistemas de Informação - São Carlos - SP 2018

Exercício 1:

**Compare os modelos sequencial (clássico, cascata) e prototipação:**

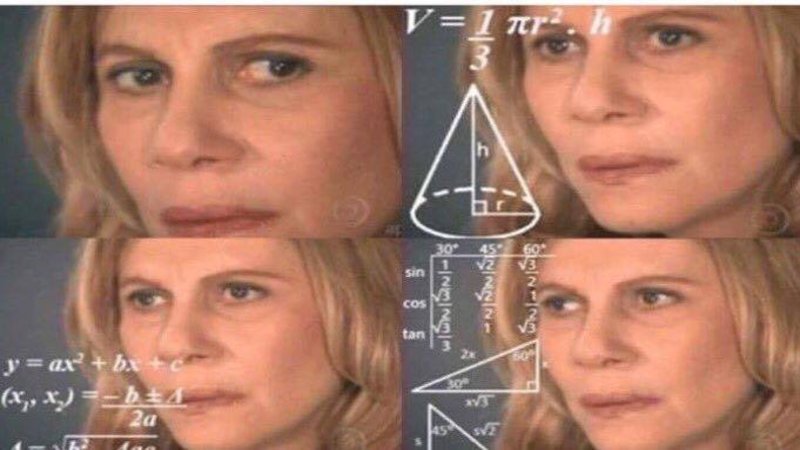
* Vantagens
* Desvantagens
* Problemas que apresenta

**Cascata**

* **Vantagem Sequencial:** fácil de entender e aplicar.
* **Desvantagens:** 
  + Ociosidade entre os desenvolvedores, há muita demora para um programador entrar em ação (por exemplo).
  + Dificuldade em realizar correções, pois os erros “teoricamente” são percebidos somente em etapas tardias

**Protótipo**

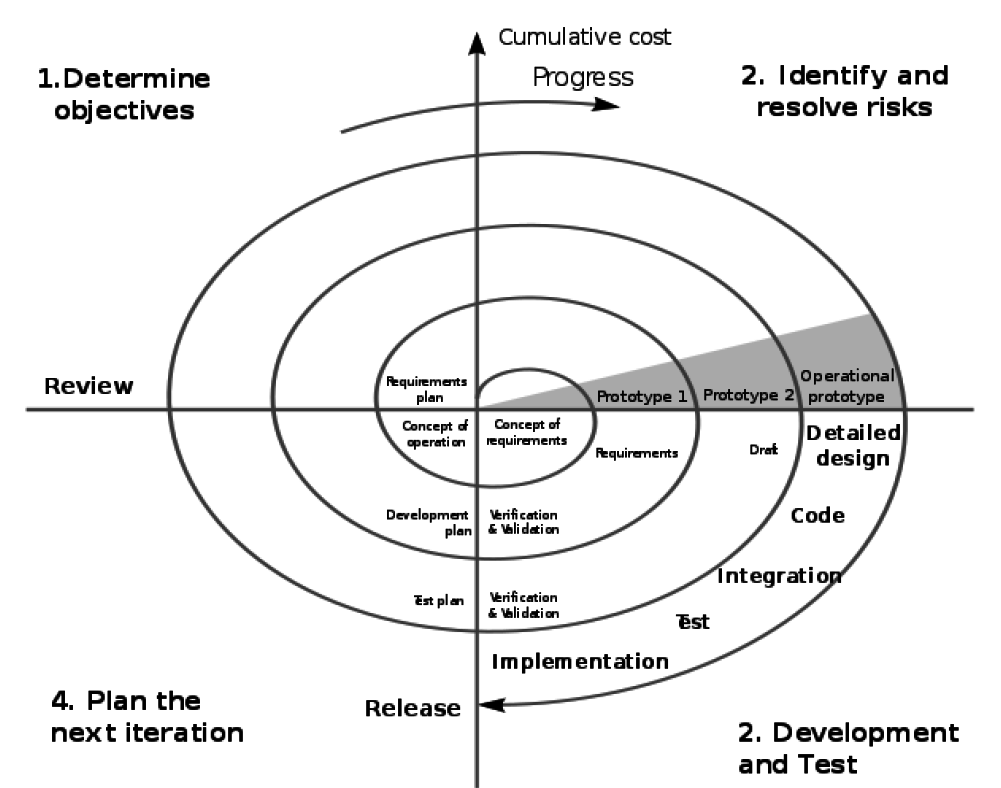
* **Vantagens:** o cliente tem ideia de como o sistema será desenvolvido; útil quando o cliente não sabe claramente quais são os requisitos; utiliza linguagens de quarta geração.
* **Desvantagens:** o protótipo não é o sistema (que deve ser construído); pode ser entregue somente a “casca” do sistema; cliente pode pedir continuamente e isso afeta o custo e tempo de desenvolvimento do projeto.



Exercício 2:

Compare os modelos de processo incremental e espiral quanto a:

* **Desvantagens**
  + Incremental e Iterativo: Essas 2 abordagens funcionam muito bem juntas ( na esmagadora maioria dos métodos ágeis) , mas separadas perdem sentido. Uma abordagem unicamente iterativa iria trabalhar em uma única funcionalidade ignorando todas as demais, até que ela ficasse ‘perfeita’, e aí sim passaria para outra funcionalidade. Uma abordagem unicamente Incremental trabalharia em diversas frentes, lançando diversas funcionalidades, mas onde nenhuma delas funciona corretamente pois não há um esquema interessante de revisão do que já foi feito. Diferente do modelo espiral, não leva em consideração os riscos que podem surgir.
  + Espiral: Exige que se tenha gerentes experientes, por suas características de avaliação e planejamento baseadas em risco. Acompanhar e controlar o andamento do projeto pode se tornar mais difícil, já que o modelo em espiral pode levar ao desenvolvimento em paralelo de múltiplas partes do projeto, cada uma sendo abordada de uma maneira diferente.
* **Vantagens** 
  + Incremental e Iterativo: O software é aperfeiçoado em vários momentos, pois em cada iteração são revistos e melhorados pontos que podem ter passado despercebidos antes, ou problemas que surgiram na iteração anterior. Graças ao processo incremental, há diminuição no tempo de entrega, já que a aplicação é dividida por funcionalidades. Isso eventualmente tem uma segunda função, que é a de deixar o sistema mais “modularizado” e com baixo nível de acoplamento e maior independência entre as funcionalidades, ao invés de criar um “sistemão monolítico” que cuida de tudo. Essa prática também tem impacto positivo em questões de performance.
  + Espiral: Capacita desenvolvedor e cliente a entender e reagir aos riscos em cada etapa do processo de desenvolvimento. Compartilha vantagens do modelo Iterativo, pois a cada volta na espiral, as atividades do desenvolvimento podem ser re-executadas com maior ou menor intensidade sobre determinados artefatos de software.



Exercício 3:

* **Quando usar RUP? Quando não usar RUP?**
  + RUP deve ser usado quando há necessidade de solucionar questões muito específicas da organização, já que é um *framework* e como tal pode ser adaptado e alterado conforme demanda. Porém é necessário avaliar em que grau o desenvolvimento em questão deve ser orientado à riscos, para poder decidir acertadamente entre RUP e espiral. Trata-se de um processo robusto e bem definido com a geração de artefatos importantes, onde os maiores riscos são atacados primeiro, diminuindo as chances de fracasso do projeto.
  + Pode se apresentar complexo e trabalhoso, não sendo recomendado para projetos de pequeno porte. Também costuma exigir mais experiência da equipe.
* **Comparação entre sequencial linear (cascata, prototipação, incremental, espiral) X RUP**
  + O RUP possui princípios de iteração e eventualmente procedimentos incrementais, porém costuma ser utilizado para processos mais robustos, devido à ser mais complexo e trabalhoso. Possui mais versatilidade em relação ao modelo cascata. Cada iteração no RUP resulta em uma versão executável. Apesar de também possuir alguma orientação à riscos, podem haver casos em que o modelo espiral se encaixe melhor. No RUP não é tão fácil identificar falhas bobas logo no início, o que é bem comum no modelo de prototipação.





**Casos de Uso**

* Baseado no documento de requisitos.
* Representa uma possível utilização do sistema por um ator, que vai interagir com o sistema alvo.
  + Atores: um sensor, um usuário, um subsistema, interação direta ou indireta.
* Narra a interação entre os atores e o sistema
* Pode ter vários **cenários** de interação entre o **sistema** e **atores**, com sucesso ou fracasso, para atingir um objetivo.
* Cada cenário e uma sequência específica de ações e interações, ou um caso particular desde o início até a conclusão.
  + Cenário
    - caixa de um banco efetua retirada de dinheiro de uma conta do tipo corrente para uma conta poupança
  + Caso de Uso
    - Retirada de Dinheiro
* R!  
  R2
* R3
* Pré e pós condições
* Cenário de sucesso principal
* fluxo alternativo
* “verbo” “para cada”
* resumido vs completo
* asbtrato - sem detalhes da implementação
* concreto - com os detalhes. imp soft/hard,
* Relacionamentos entre UC’s
  + seta pontilhada << include >>
  + “Incluir o caso de uso: Movimento Conta” - modularização
  + Relacionamento de extensão