

## **AULA 02 – Processos de Software**

### **Introdução:**

Em nossa segunda aula, trabalharemos o conhecimento sobre processos de desenvolvimento de software. Compreenderemos o motivo pelo qual sempre partimos do modelo clássico para chegarmos aos modelos mais utilizados na atualidade, os métodos ágeis.

### **Contextualizando:**

Como aprendemos na primeira aula, a engenharia de software surge com a finalidade de tornar o desenvolvimento de software algo mais sistematizado, utilizando metodologias, técnicas e ferramentas. Vimos que outrora o software era desenvolvido de forma artesanal e isto gerou problemas sérios a empresas e gerentes de projetos. Uma das formas de iniciarmos um desenvolvimento que possa ser planejado, controlado, executado e testado é com a adoção de um processo de desenvolvimento de software. Assim como todo processo de fabricação de produtos na indústria, o software também requer um modelo de processo.

## **PROCESSOS DE DESENVOLVIMENTO DE SOFTWARE**

A construção de software é um processo de aprendizado social e iterativo sendo o resultado o capital denominado software. Este capital intelectual incorpora conhecimento coletado, filtrado e organizado. Todo modelo de processo preocupa-se em descrever uma metodologia a ser seguida, composta por atividades de apoio, atividades metodológicas, atividades (ações) e tarefas. Veja a figura 1.



Figura 1: Estrutura básica de uma metodologia de processo

Um modelo de processo também pode ser definido como um arcabouço que contenha tarefas necessárias para a construção de software com um bom padrão de qualidade. Um processo define a abordagem adotada para elaboração do software.

Então, o que ganhamos ao adotar um modelo de processo?

1. Estabilidade durante o processo de desenvolvimento
2. Maior controle sobre o processo
3. Organização do processo
4. Agilidade
5. Sem dúvidas, a documentação que é escrita e mantida ao longo do processo
6. E finalmente, adotamos um processo que seja aderente ao perfil profissional que temos na organização.

Os modelos de processos são constituídos de fluxos de processo que podem ser linear (figura 2), iterativo (figura 3), evolucionário (figura 4) e paralelo (figura 5) ou uma combinação de dois ou mais fluxos de processo.



Figura 2: Fluxo de processo linear

Os fluxos de processo linear necessariamente seguem uma ação após a outra até o final do modelo.

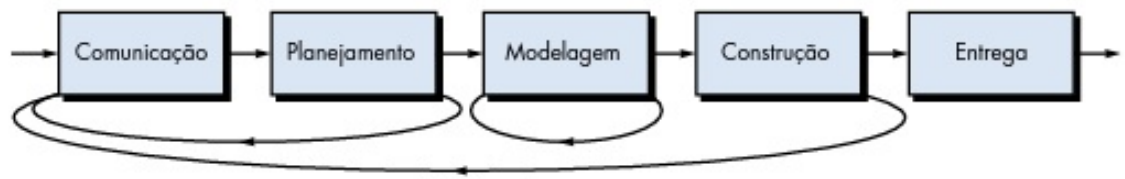


Figura 3: Fluxo de processo iterativo

Os fluxos de processo iterativos possuem um formato de retorno dentro de uma mesma atividade ou mais. Mesmo que haja uma sequência a ser seguida, é permitida a iteratividade entre atividades e tarefas.

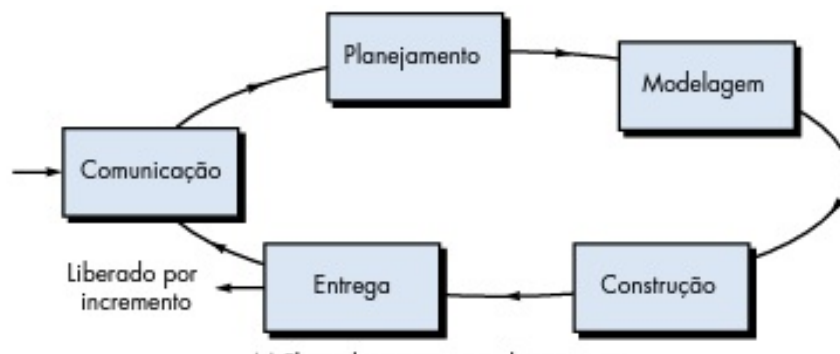


Figura 4: Fluxo de processo evolucionário

Os fluxos de processo do tipo evolucionário possuem um aspecto de iteratividade entre todas as atividades e tarefas dentro de um curto espaço de tempo. Sendo que as mesmas atividades e tarefas são revistas durante o incremento do desenvolvimento.

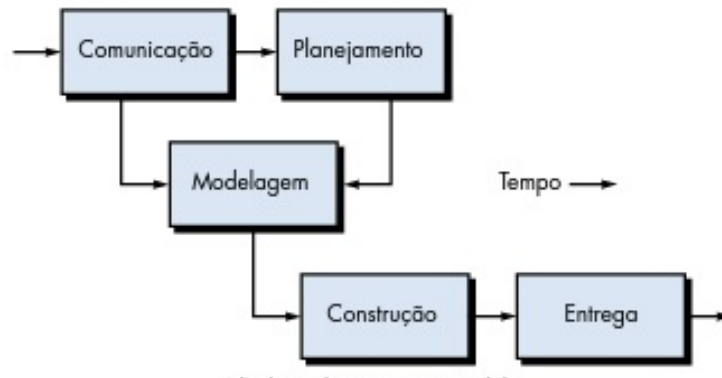


Figura 5: Fluxo de processo paralelo

Os fluxos de processo em paralelo permitem que várias pequenas unidades do software sejam desenvolvidas em paralelo e independente.

### **Modelo de Processo Clássico**

Este modelo, também conhecido por modelo cascata, é uma abordagem sequencial e sistemática das atividades e tarefas. Permitindo que o resultado seja visto somente no final do projeto. Este paradigma é o mais antigo da engenharia de software. Sua eficácia é questionável, visto que projetos que o utilizaram, raramente conseguiram seguir o fluxo sequencial em sua plenitude. Inevitavelmente, projetos que se utilizaram deste modelo, não obtiveram um resultado ideal. Um dos grandes problemas para este modelo encontra-se na coleta dos requisitos. É muito difícil compreendermos e obtermos 100% dos requisitos necessários para o desenvolvimento de um software apenas em um momento do projeto. A figura 6 exemplifica o modelo em questão.

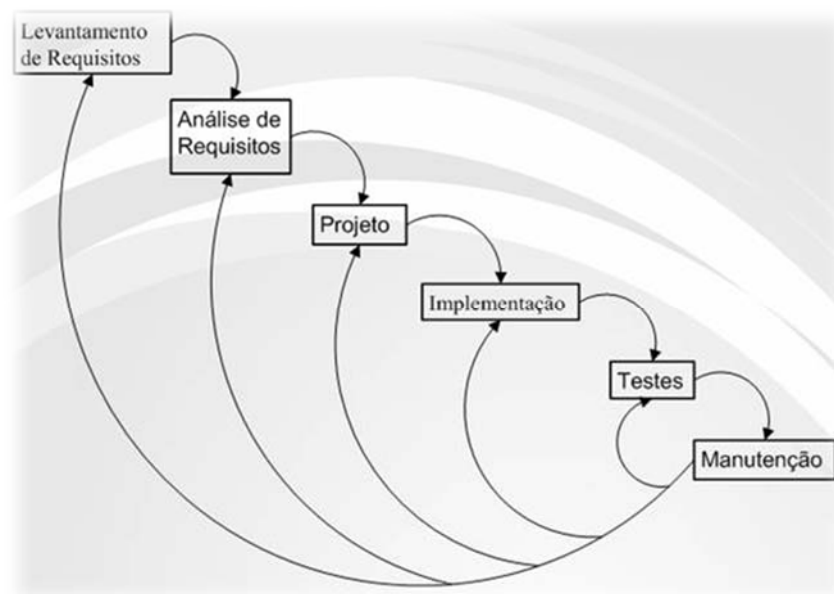


Figura 6: Modelo Clássico

### Modelo de Processo Incremental

O modelo incremental refina-se a cada coleta de requisitos. A cada nova elicitação de requisitos o projeto expande-se. Este modelo combina fluxos de processo lineares e paralelos. É um modelo sequencial de forma escalonada em acordo com o avanço do tempo de execução do projeto. A figura 7 exemplifica o modelo em questão.

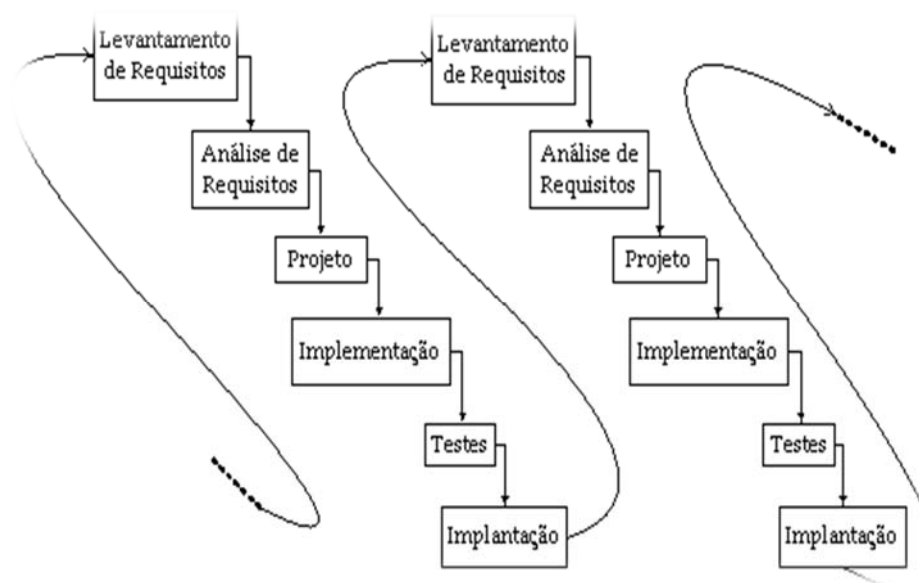


Figura 7: Modelo Incremental

Este modelo é útil quando não há pessoal disponível para implementação completa do software. Os primeiros incrementos podem ser feitos com poucos profissionais. Também podem ser desenvolvidos para outra plataforma de hardware e de software com menor capacidade.

### **Modelo de Processo Evolucionário**

Este modelo aplica-se a sistemas complexos que evoluem ao longo do tempo de desenvolvimento. As necessidades de negócio e de produto mudam com muita frequência. Os prazos geralmente são apertados e são produzidos para mercados muito específicos. Há dois tipos de modelos evolucionários: Prototipação e Espiral.

O modelo Prototipação (figura 8) é um paradigma muito utilizado dentro do contexto de Interação Humano-Computador. Os requisitos geralmente são obscuros e difíceis de serem elicitados. O primeiro protótipo pode servir como um primeiro sistema em uso, ou pode ser descartável.

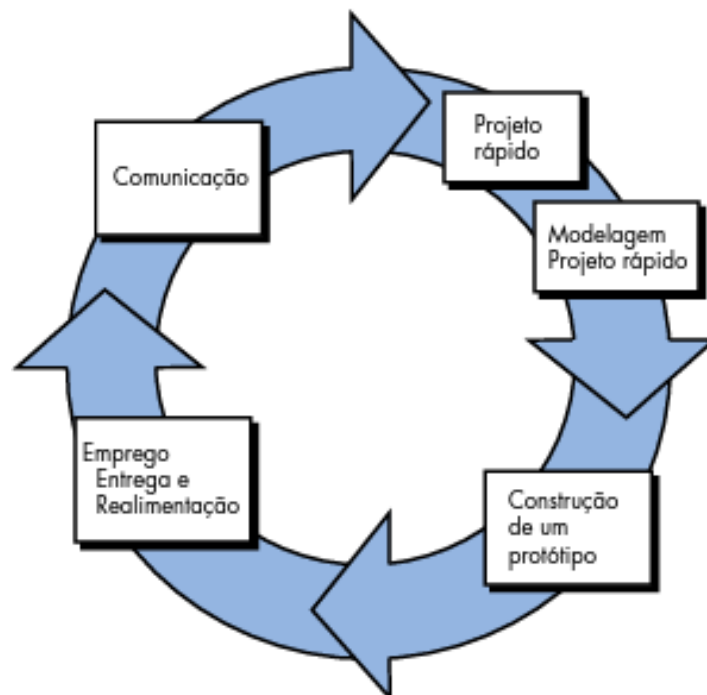


Figura 8: Modelo evolucionário: prototipação

Já o modelo espiral (figura 9) possui a natureza iterativa da prototipação em conjunto com aspectos sistemáticos do modelo clássico. Ele foi desenvolvido para

modelos de processos dirigidos a riscos. Sua abordagem cíclica é voltada para a ampliação e incremento do grau de definição inicial. Ele diminui o grau de risco do projeto com pontos âncoras de controle. Evolui somente após passar por uma série de iterações.

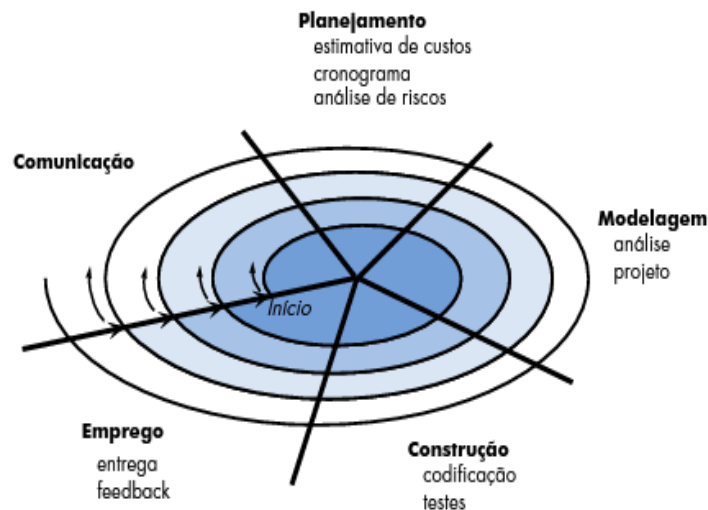


Figura 9: Modelo evolucionário: espiral

O modelo concorrente (figura 10) representa elementos concorrentes agregando conceitos do modelo iterativo.

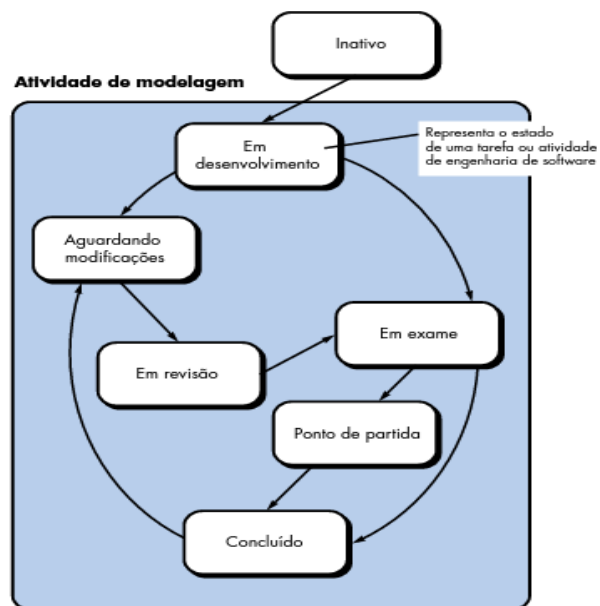


Figura 10: Modelo evolucionário: concorrente

## Modelo de Processo Especializado

Os modelos especializados baseiam-se em componentes, métodos formais e orientação a aspectos. Estes modelos incorporam características do modelo espiral, com abordagem iterativa e reuso de componentes. Os métodos formais são utilizados para especificação forma de modelos matemáticos, que possuem uma notação rigorosa. Modelos que envolvem alto custo, poucos desenvolvedores e um treinamento extensivo. O modelo orientado a aspectos oferece uma abordagem metodológica para definir, especificar, projetar e construir aspectos. Possui mecanismos como sub-rotinas e herança para localização da expressão de uma restrição cruzada. É um tipo de processo não maduro ainda. Sua natureza é paralela e concorrente.

## Modelo de Processo RUP

A origem deste modelo encontra-se nos primeiros trabalhos de Jacobson, Grady Booch e James Rumbaugh. Ele foi constituído para ser dirigido por casos de uso (diagramas da UML), centrado na arquitetura, iterativo e incremental. Possui recursos melhores que os modelos tradicionais, princípios de desenvolvimento ágil, comunicação com cliente e métodos racionalizados. A figura 10 exibe o cruzamento das quatro principais fases com as disciplinas observadas durante o modelo.

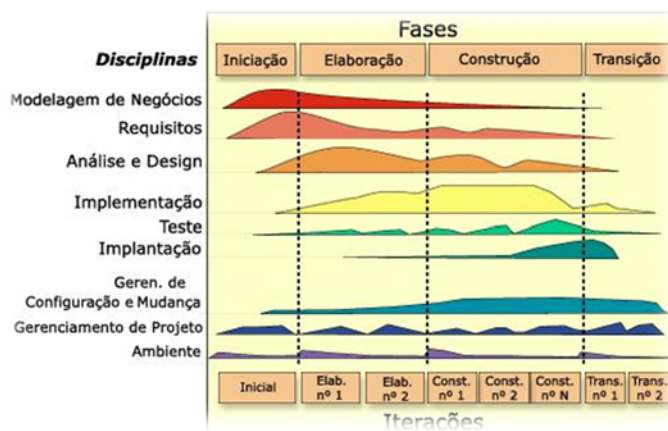


Figura 10: RUP



## **Métodos Ágeis**

Foram idealizados para garantir agilidade, resposta apropriada a mudanças (equipes e tecnológicas), colaboração, menor custo nas mudanças. Outra característica é ser adaptável e capaz de auxiliar na administração da imprevisibilidade.

Princípios dos métodos ágeis:

1. Satisfazer o cliente
2. Pedidos de alteração
3. Entrega rápida
4. Comercial e desenvolvimento trabalhando juntos
5. Time motivado
6. Comunicação efetiva
7. Software funcionando
8. Desenvolvimento sustentável
9. Excelência técnica
10. Simplicidade
11. Melhores arquiteturas e requisitos
12. Feedback e avaliação.

Há uma variedade de métodos ágeis, entre eles:

1. XP – extreme programming
2. DAS – desenvolvimento adaptativo de software
3. DSDM – método de desenvolvimento dinâmico de sistemas
4. Scrum
5. Crystal
6. FDD – desenvolvimento guiado por características
7. LSD – enxuto
8. AM – modelo ágil
9. AUP – unificado ágil.

Os métodos ágeis surgem com a proposta de serem utilizados em projetos menores, no entanto, seu sucesso atingiu projetos maiores.

### **Pesquisa**

Busque informações detalhadas sobre os métodos ágeis listados nesta lição. Tente montar um cruzamento de características existentes em cada um e compará-los analiticamente.

### **Trocando Ideias**

Os modelos de processos existem para nos orientar no processo de desenvolvimento de projetos, jamais nos atrapalhar. Adotamos um modelo de processo conforme nossas necessidades. Como sempre falo, menos é mais. O mínimo que você adotar dos conceitos de um ou mais modelos de processo irá auxiliá-lo na obtenção de melhores resultados, ou seja, o software aceito com um bom grau de satisfação por nossos stakeholders.

### **Síntese**

Nessa aula comparamos vários modelos de processo. Compreendemos a necessidade estudarmos o modelo clássico, mesmo que o mesmo não tenha sido um caso de sucesso. Ele foi o início de toda uma preocupação de formalização dos processos de desenvolvimento de software.

### **Compartilhando**

O modelo de processo de desenvolvimento de software, conforme é implementado, auxilia em todas as fases do desenvolvimento. Auxilia na implantação de sistemas de qualidade, auxilia na garantia de um processo e software com excelência.