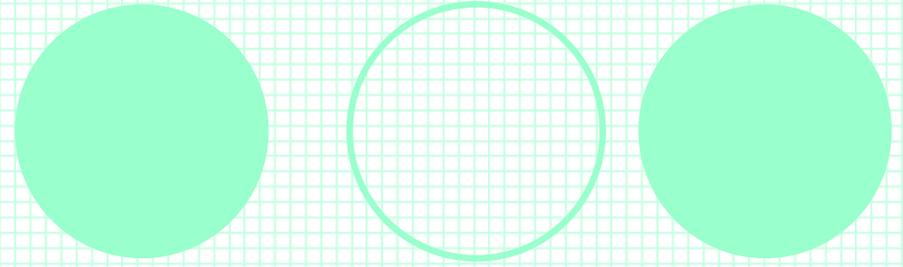




# **Comparativo entre Processos Ágeis**

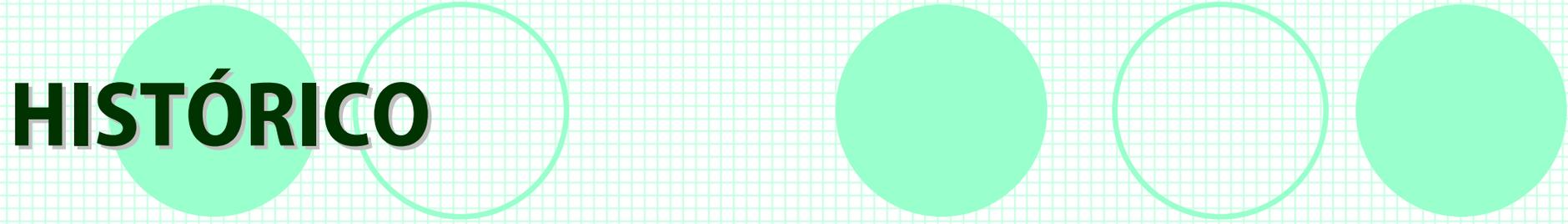
Daniel Ferreira  
dfs3@cin.ufpe.br

# O que discutiremos:



- Histórico
- Os Princípios Ágeis
- Comparação
  - Do ponto de vista incremental
  - Do ponto de vista funcional
- Vantagens e Desvantagens
- Conclusões

# HISTÓRICO



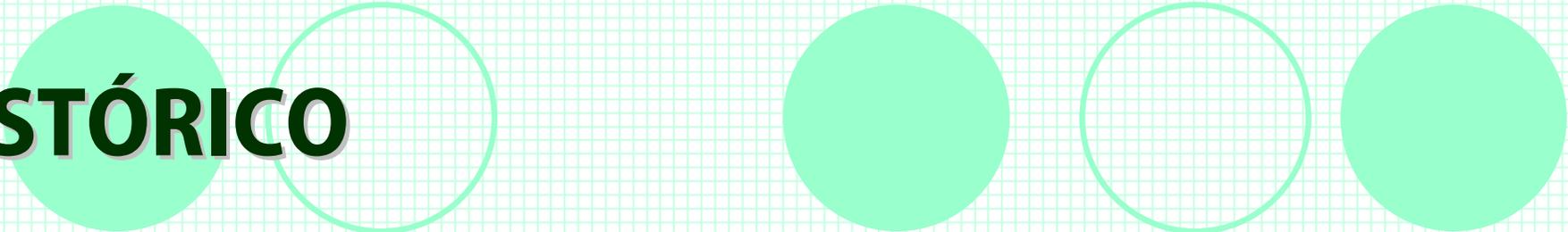
- A indústria de software sempre contou com métodos cujos processos de desenvolvimento eram baseados em um conjunto de atividades predefinidas, descrita como processos.

**[AMBLER, 2004]**

- Quase sempre era exigido um levantamento completo de requisitos, seguido por um projeto de alto-nível, de uma implementação, de uma validação e, por fim, de uma manutenção.

**[SOMMERVILLE, 2003]**

# HISTÓRICO



- Métodos ágeis começaram a surgir na segunda metade de **1990**.
- Descontentamento com métodos atuais devido a **extensa documentação** exigida.
- Em **fevereiro de 2001** foi publicado o Manifesto Ágil

# DESENVOLVIMENTO ÁGIL DE SOFTWARE

“Agile software development is a style of software development characterized by an emphasis on **people, communication, working software**, and responding to change.”

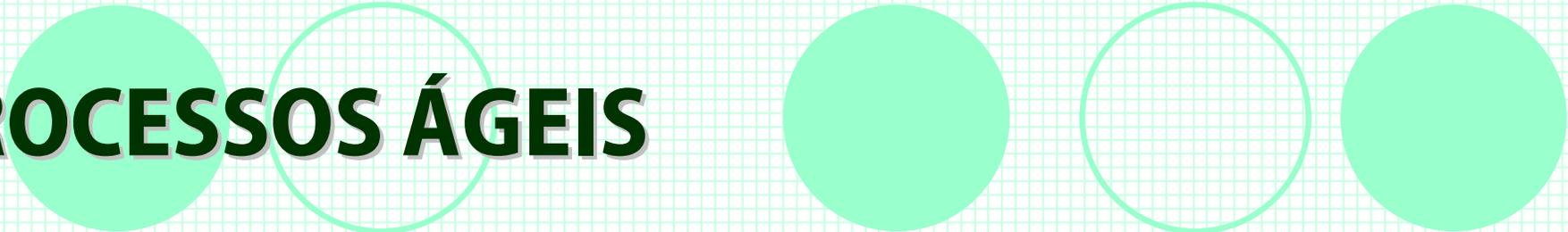
**[BECK, et. al., 01]**

# OS PRINCÍPIOS ÁGEIS

- Do total dos 12 princípios inicialmente propostos, podemos destacar:
  - Prioridade é satisfazer o cliente com entregas contínuas e frequentes;
  - Equipes de negócio e desenvolvimento devem sempre trabalhar juntas;
  - Melhor maneira da informação circular é através de uma conversa face-a-face;
  - Indivíduos e interações entre eles são mais importantes que processos e ferramentas.
  - Simplicidade é essencial

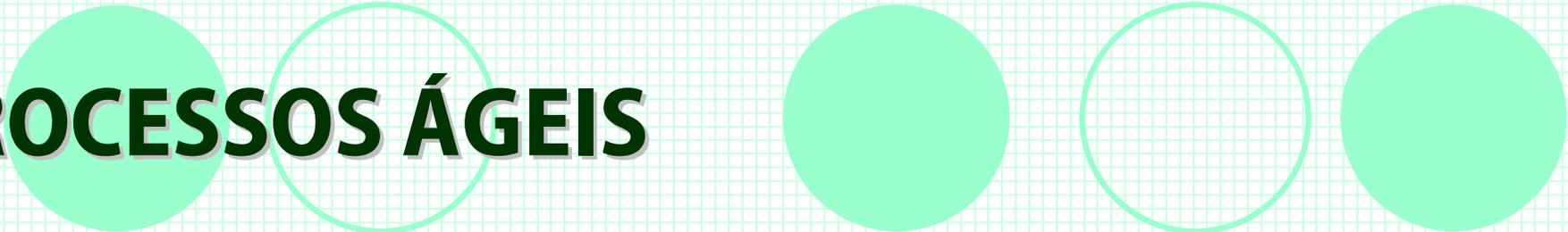
**[BECK, et. al., 01]**

# PROCESSOS ÁGEIS



- Scrum
  - Defende o desenvolvimento iterativo e incremental de software
- XP
  - Abordagem deliberada e disciplinada
- Crystal
  - Família de metodologias que varia de acordo com complexidade e tamanho do projeto
- Agile RUP
  - Metodologia que preza pela agilidade mas com documentação rica baseada no RUP.
- FDD
  - Metodologia centrada na modelagem do domínio da aplicação.

# PROCESSOS ÁGEIS



- XP

- Abordagem deliberada e disciplinada

- Crystal

- Família de metodologias que varia de acordo com complexidade e tamanho do projeto

- Agile RUP

- Metodologia que preza pela agilidade mas com documentação rica baseada no RUP.

# PROCESSOS ÁGEIS

**eXtremme  
Programming**

**Scrum**

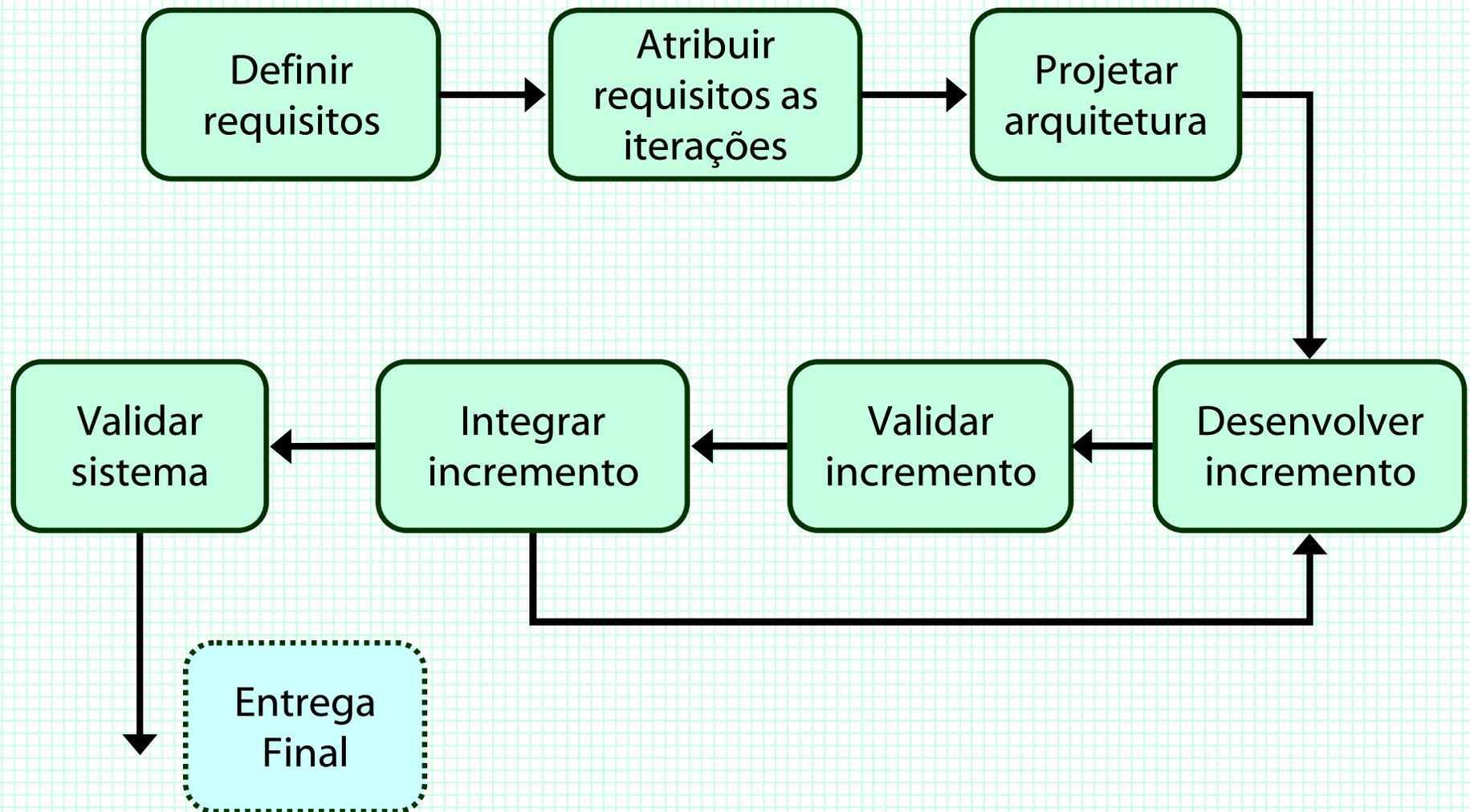
**Feature Driven  
Development**

**Adaptive Software  
Development**

**Crystal Clear**

**Agile RUP**

# DO PONTO DE VISTA INCREMENTAL



# DEFINIR REQUISITOS

## ● XP

- User stories: Descrições sucintas a respeito das funcionalidades do sistema

## ● Scrum

- Requisitos são listados gerando o Product Backlog, agrupados de acordo com a prioridade.

## ● FDD

- Uso também de user stories com o auxílio e diagramas UML de classe e de sequência.

## ● ASD

- Nenhuma atividade explícita, mas são realizadas sessões JAD para definição dos requisitos

# ATRIBUIR REQUISITOS AS ITERAÇÕES

- XP
  - Equipe técnica e clientes definem as user stories que serão desenvolvidas nas iterações. As iterações duram de 1 a 4 semanas.
- Scrum
  - Definição do Sprint Backlog. As Sprints (iterações) duram no máximo 30 dias.
- FDD
  - As características são agrupadas, priorizadas e distribuídas aos responsáveis pela seu desenvolvimento. Iterações de no máximo 2 semanas.
- ASD
  - Definição do número de iterações e quais requisitos cada um terá. As iterações duram de 4 a 8 semanas.

# PROJETAR ARQUITETURA

## ● XP

- Deve ser feito paralelamente a criação das user stories, mas sem especificar técnica ou artefatos.

## ● Scrum

- Projeto geral do sistema baseado no Product Backlog mas também sem especificar técnicas

## ● FDD

- Criação de diagrama de classes UML para representar a arquitetura do sistema

## ● ASD

- Não sugere nenhuma atividade relacionada a esta etapa.

# DESENVOLVER INCREMENTO

## ● XP

- Realizada durante as iterações, para cada conjunto de user stories selecionadas.

## ● Scrum

- Realizada durante a Sprint, para cada Sprint Backlog definido.
- Durante cada Sprint são realizadas reuniões diárias de 15-30 minutos para todos os envolvidos se manterem informados.

# DESENVOLVER INCREMENTO

## ● FDD

- Análise da documentação existente, geração de Diag. de Seqüência da UML, refinamento do modelo gerado nas atividades anteriores e implementação das características que serão desenvolvidas durante a iteração corrente.

## ● ASD

- Implementação dos requisitos que fazem parte da iteração corrente.

# VALIDAR INCREMENTO

- XP

- Execução de testes de aceitação e testes de unidade

- Scrum

- Atividade de validação realizada no final da Sprint.

- FDD

- Os testes e inspeções são executados pelos próprios programadores após a implementação.

- ASD

- São verificadas a qualidade técnica e funcional do sistema, onde um par de programadores é responsável por revisar e avaliar o código do sistema.

# INTEGRAR INCREMENTO

## ● XP

- À medida que o código vai sendo gerado ele vai sendo integrado, evitando problemas de compatibilidade cedo.

## ● Scrum

- A integração dos resultados das várias Sprints acontece ao final de cada uma delas.

## ● FDD

- Integração ao final de cada iteração, após os testes e inspeções.

## ● ASD

- Não sugere nenhuma atividade específica de integração dos incrementos

# VALIDAR SISTEMA

- XP

- O sistema é disponibilizado ao cliente para que o mesmo realize validações;

- Scrum

- O cliente valida o sistema integrado em uma reunião no último dia da Sprint, chamada **Revisão da Sprint**. Os participantes avaliam o incremento e decidem sobre as atividades seguintes.

- FDD

- Esta atividade ocorre através das inspeções e dos testes de integração

- ASD

- Não sugere nenhuma atividade específica de validação do sistema como um todo.

# ENTREGA FINAL

## ● XP

- Ocorre quando o cliente estiver satisfeito com o sistema e não tiver mais nada a acrescentar as funcionalidades

## ● Scrum

- Ocorre quando não existirem mais itens no Product Backlog

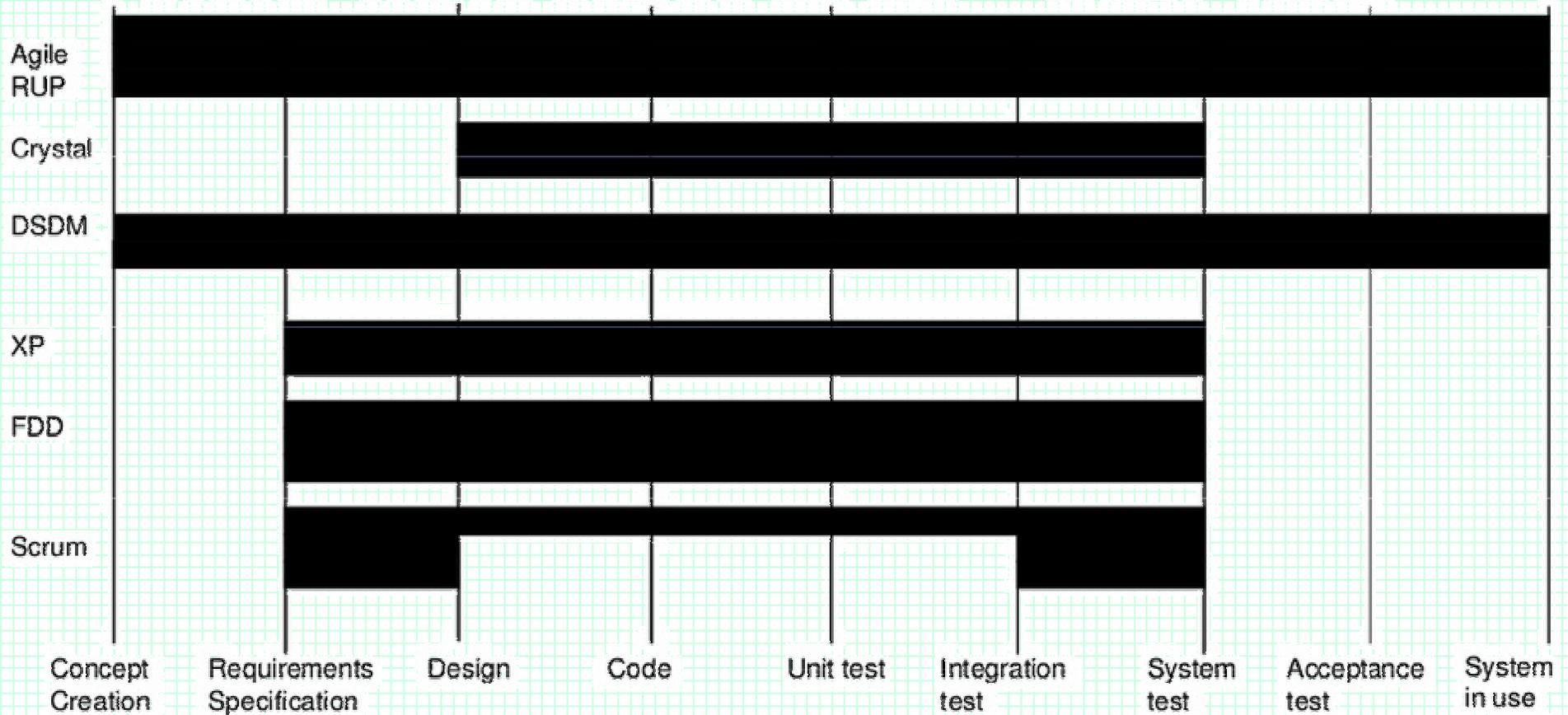
## ● FDD

- Todos os conjuntos de características implementados

## ● ASD

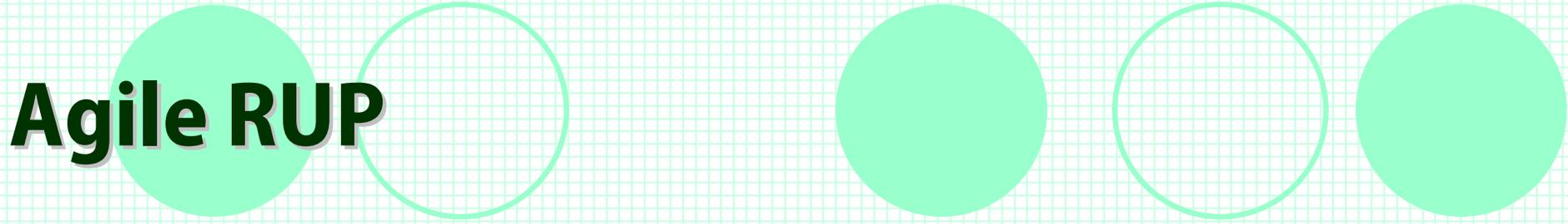
- Todos os requisitos desenvolvidos

# DO PONTO DE VISTA FUNCIONAL



[PRABHAKARAN, 09]

# Agile RUP



- Fases

- Início, Elaboração, Construção, Transição

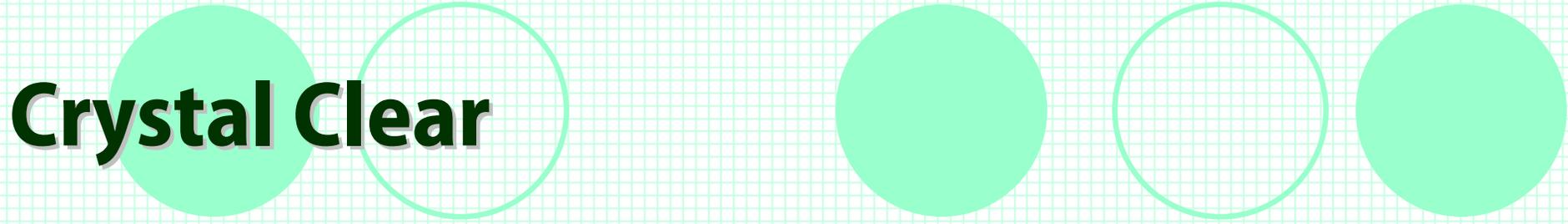
- Áreas

- Modelagem, implementação, testes, desenvolvimento, gerenciamento de configuração, gerenciamento de projeto, ambiente

- Filosofias

- Sua equipe sabe o que estão fazendo; Simplicidade; Agilidade; Foco em atividades de maior valor; Independência de ferramentas.

# Crystal Clear



- Propriedades

- Entregas frequentes de código usável; Melhorias reflexivas; Comunicação osmótica.

- Estratégias

- Arquitetura incremental; Radiadores de informação

- Técnicas

- Standup meetings diárias; Programação em pares; Gráficos “burn”.

# eXtreme Programming

- Valores

- Comunicação, simplicidade, feedback, coragem e respeito.

- Atividades

- Codificar, testar, ouvir, modelar

- Práticas

- Programação em pares; Planejamento com “games”; TDD; Equipe completa; Integração contínua; Melhoria de modelagem; Releases pequenas; Padrões de codificação; Modelagem simples; ritmo sustentável.

# Feature Driven Development

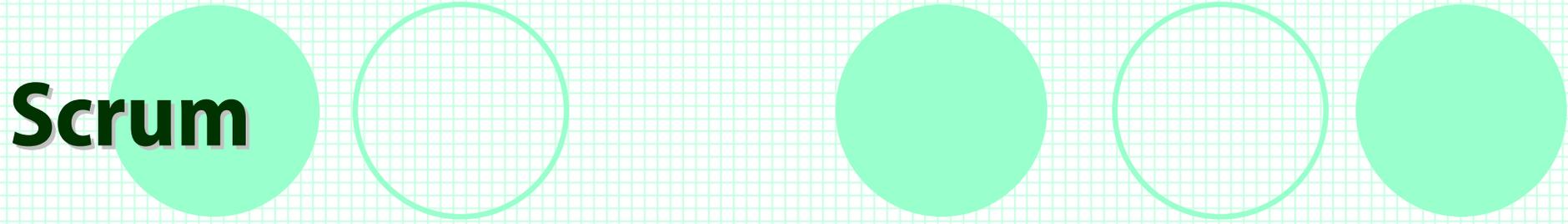
## ● Atividades

- Desenvolver modelo global; Construir lista de funcionalidades; Planejamento, modelagem e desenvolvimento baseados em funcionalidades; Marcos

## ● Boas práticas

- Modelagem de objetos de domínio
- Desenvolvimento por funcionalidade
- Times por funcionalidades
- Inspeções
- Gerência de Configuração
- Builds regulares
- Visibilidade do progresso e resultados

# Scrum



## ● Técnicas

- Criação de times
- Criação de Backlogs
- Segmentação do projeto
- Reuniões diárias
- Gráficos "burn"

## ● Fases

- Revisão de plano de release
- Distribuição, revisão e ajustes nos padrões do produto
- Sprint
- Sprint review
- Fechamento

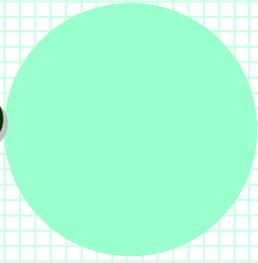
Características	XP	Scrum	FDD	ASD	Crystal
Abordagem	Incrementos iterativos	Incrementos iterativos	Iterativo	Iterativo	Incremental
Tempo de iteração	1-6 semanas	2-4 semanas	2 dias a 2 semanas	4-8 semanas	Depende do método
Tamanho do time	Pequenos < 20	Qualquer tamanho	Médios/ Grandes	Pequenos 5-9	Qualquer tamanho
Comunicação	Informal	Informal	Documentos	Informal	Informal
Tamanho do projeto	Pequenos	Qualquer tamanho	Projetos complexos	Pequenos	Qualquer tamanho
Envolvimento do cliente	Envolvido	Envolvido como PO	Envolvido através de relatórios	Envolvido através das releases	Envolvido através das releases
Documentação	Básica	Básica	Documentação é importante	Básica	Básica
Especialidades	TDD, user stories, refactoring	Sprint, product backlog, Planning poker, scrum master	Diagramas UML	Ciclos de aprendizado	Métodos adaptativos

master



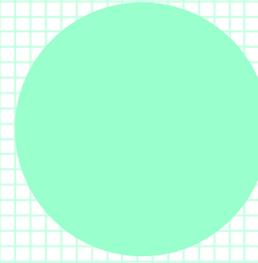
# **VANTAGENS E DESVANTAGENS**

# XP



## VANTAGENS 😊

- Práticas técnicas fortes
- Cliente determina as prioridades. Desenvolvedores determinam as estimativas.
- Várias oportunidades de feedback
- Técnica ágil mais adotada mundialmente.



## DESVANTAGENS ☹️

- Requer cliente sempre presente
- Documentação se baseia apenas no código e comunicação verbal.
- Dificuldade para novas equipes adotarem as práticas num projeto já existente.

# SCRUM

## VANTAGENS 😊

- Complementa práticas existentes
- Times auto-organizados e com feedback
- Participação e direção do cliente
- Prioridades baseadas no valor para o cliente
- Única abordagem que possui certificação.

## DESVANTAGENS ☹️

- Provê apenas suporte para gerenciamento de projetos
- Não especifica técnicas práticas
- Tempo necessário para o cliente definir as prioridades para cada requisito.

# FDD

## VANTAGENS 😊

- Vários times podem trabalhar em paralelo
- Todos os aspectos do projeto são rastreados por funcionalidade
- Modelagem e desenvolvimento por funcionalidade são fáceis de entender e de adotar.
- Escalável para times e projetos grandes

## DESVANTAGENS ☹️

- Iterações não são tão bem definidas como nas outras metodologias
- Promove “donos” do código

# AGILE RUP

## VANTAGENS 😊

- Metodologia robusta com muitos artefatos e disciplinas cobertas
- Escalável
- Documentação ajuda a comunicação ser distribuída em diferentes ambientes
- Prioridades baseadas no risco (de negócio ou técnico)

## DESVANTAGENS ☹️

- Muita documentação pode ser prejudicial para projetos pequenos
- Pouca atenção para dinâmicas do time
- Documentação bem mais formal que outras metodologias

# CRYSTAL

## VANTAGENS 😊

- Metodologia escalável de acordo com o tamanho do projeto e sua criticidade
- Único método que especifica responsável pelo ciclo de vida de projetos críticos.
- Ênfase na parte de testes (geralmente 1 tester para cada time)

## DESVANTAGENS ☹️

- Muita documentação pode ser prejudicial para projetos pequenos
- Pouca atenção para dinâmicas do time
- Documentação bem mais formal que outras metodologias

# CONCLUSÕES

XP	Simplicidade
Scrum	Valor de negócio
FDD	Modelo de negócio
Agile RUP	Gerenciamento de Riscos
Crystal	Tamanho e complexidade

# Referências

[SOMMERVILLE, 03]

SOMMERVILLE, I. **Engenharia de Software**. São Paulo: Addison-Wesley, 2003.

[AMBLER, 04]

AMBLER, S. **Modelagem Ágil. Práticas Eficazes para a Programação Extrema e o Processo Unificado**. Porto Alegre. Bookman, 2004.

[HIGHSMITH, 02]

HIGHSMITH, J. **Agile Software Development Ecosystems**. Boston. Addison Wesley, 2002.

[ABRAHAMSSON, 02]

ABRAHAMSSON, P., Salo, O., RONKAINEN, J., & WARSTA, J. **Agile Software Development Methods: Review and Analysis**. Espoo 2002. VTT Publications 478, 2002.

[BECK, 00]

BECK, K. **Extreme Programming Explained**. Massachusetts. Addison Wesley, 2000.

[BECK, et. al., 01]

BECK, K. et. al. **Agile Manifesto**. 2001. Disponível em <<http://www.agilemanifesto.org>>. Acesso em: 31 out 2012

[LUCA, 02]

DE LUCA, J. **Feature-Driven Development (FDD) Overview Presentation**. 2002. Disponível em: <<http://www.nebulon.com/articles/fdd/download/fddoverview.pdf>>. Acesso em 25 out 2012

# Referências

[FAGUNDES, 08]

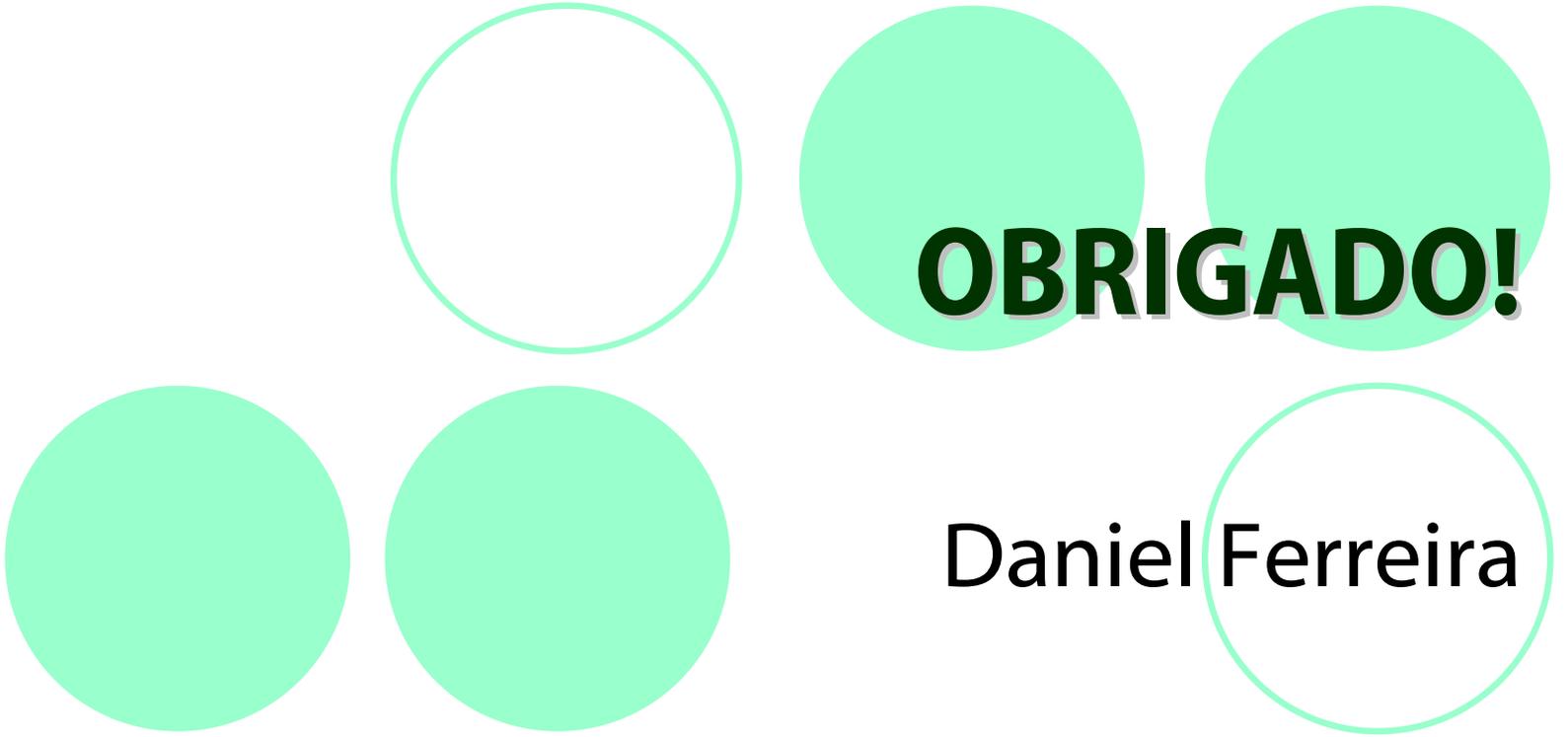
FAGUNDES, Priscila Basto, DETERS, Janice Inês, SANTOS, Sandro da Silva dos. Comparação Entre os Processos dos Métodos Ágeis: XP, SCRUM, FDD e ASD em Relação ao Desenvolvimento Iterativo Incremental. Florianópolis. 2008.

[PRABHAKARAN, 09]

PRABHAKARAN, Prasad. Attempt to Compare Agile Methods. Mahindra Satyam. 2009. Disponível em <[http://www.slideshare.net/prasad\\_keral/agile-methods-overview-compatibility-mode](http://www.slideshare.net/prasad_keral/agile-methods-overview-compatibility-mode)>. Acesso em 28 out 2012.

[AGILE, 12]

AGILE ONLY. Comparison of Agile Methodologies. Disponível em <<http://agile-only.com/master-thesis/software-dm/agile-s-dm/c-of-am>>. Acesso em 25 out 2012.



**OBRIGADO!**

Daniel Ferreira