

Gestão do Processo de Software

Modelos do Processo de Desenvolvimento (Métodos Ágeis)

Pedro Abreu Ribeiro

pmgar@dsi.uminho.pt

Departamento de Sistemas de Informação
Universidade do Minho

Gestão de Projetos de Sistemas de Informação

→ Modelos para o processo de desenvolvimento

Objectivo:

- ◆ Conhecer os modelos existentes para O Processo de Desenvolvimento .
- ◆ Dada uma situação particular de desenvolvimento de um sistema de software, saber **selecionar, combinar e aplicar** estes modelos.

Gestão de Projetos de Sistemas de Informação

O Processo de Desenvolvimento

→ Modelos do Processo de Desenvolvimento

Abordam a problemática da organização do trabalho necessário à construção (concepção, implementação e instalação) do sistema de software, em face das circunstâncias específicas.

◆ Modelos de Qualidade

Abordam a problemática da garantia da qualidade dos sistemas de software durante a sua construção.

Modelos do Processo de Desenvolvimento (Métodos Ágeis)

◆ Enquadramento

◆ Modelos não Evolutivos

- ❖ Modelo em cascata (“*waterfall*”)
- ❖ Ciclo de Vida por Fases e Atividades
- ❖ V-Model

◆ Modelos Evolutivos

- ❖ Desenvolvimento incremental
- ❖ Desenvolvimento em espiral
- ❖ Unified Process

◆ Métodos Ágeis

O Processo de Desenvolvimento

Enquadramento

- Algumas questões que deverão ter resposta em qualquer projeto:
 - Porque o projeto vai ser desenvolvido?
 - Quem está envolvido?
 - O que vai resultar do projeto?
 - Que recursos serão necessários?
 - Como serão distribuídas as responsabilidades?
 - **Que atividades serão desenvolvidas para produzir os resultados esperados?**
 - **Como serão organizadas as atividades?**
 - ...

O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

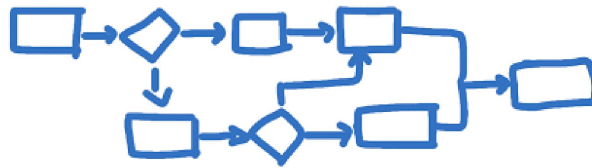
– De um modo geral, designa uma sequência de atividades, normalmente agrupadas em fases, que são executadas de forma sistemática e uniformizada por intervenientes com responsabilidades bem definidas e que, a partir de um conjunto de inputs, produzem um conjunto de outputs

O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

A software process is a coherent set of related activities that leads to the production of a software product [Sommerville 2015]



O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

– A process is a collection of activities, actions, and tasks that are performed when some work product is to be created [Pressman and Maxim 2014]

- ❖ An activity strives to achieve a broad objective (e.g., communication with stakeholders) and is applied regardless of the application domain, size of the project, complexity of the effort, or degree of rigor with which software engineering is to be applied.
- ❖ An action (e.g., architectural design) encompasses a set of tasks that produce a major work product (e.g., an architectural design model).
- ❖ A task focuses on a small, but well-defined objective (e.g., conducting a unit test) that produces a tangible outcome.

O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

– In the context of software engineering, a process is not a rigid prescription for how to build computer software [Pressman and Maxim 2014]

- ❖ Rather, it is an adaptable approach that enables the people doing the work (the software team) to pick and choose the appropriate set of work actions and tasks.
- ❖ The intent is always to deliver software in a timely manner and with sufficient quality to satisfy those who have sponsored its creation and those who will use it.

O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

– There are many different software processes but all must include four activities that are fundamental to software engineering [Sommerville 2015]

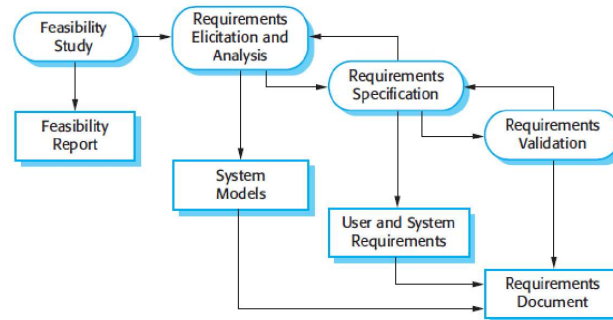
- ❖ Software specification
- ❖ Software design and implementation
- ❖ Software validation
- ❖ Software evolution

O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

– Software specification

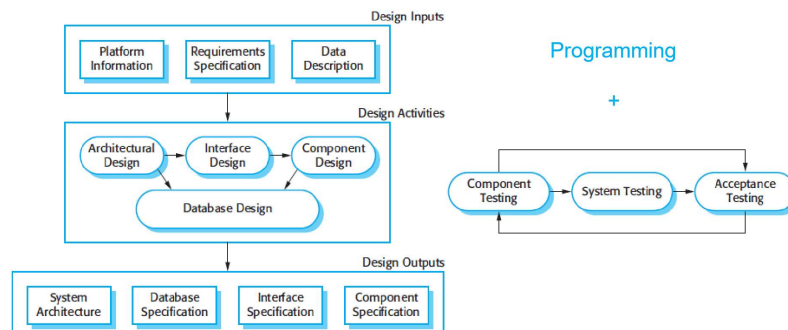


O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

– Software design and implementation

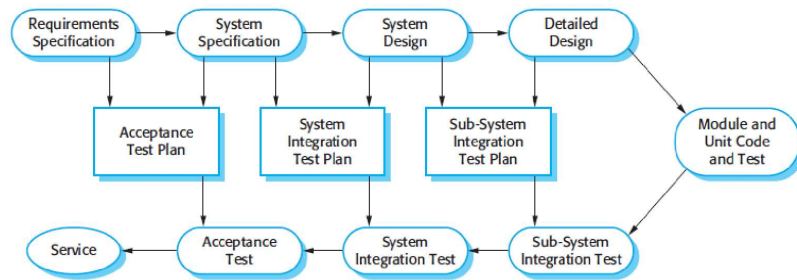


O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

– Software validation

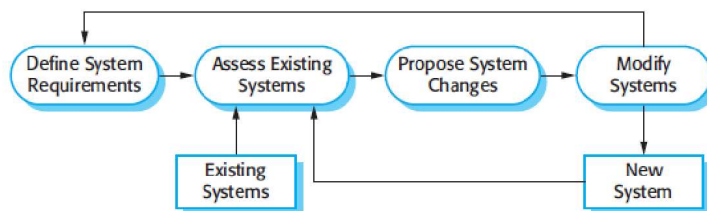


O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

– Software evolution



O Processo de Desenvolvimento

Conceito de Processo

◆ Processo

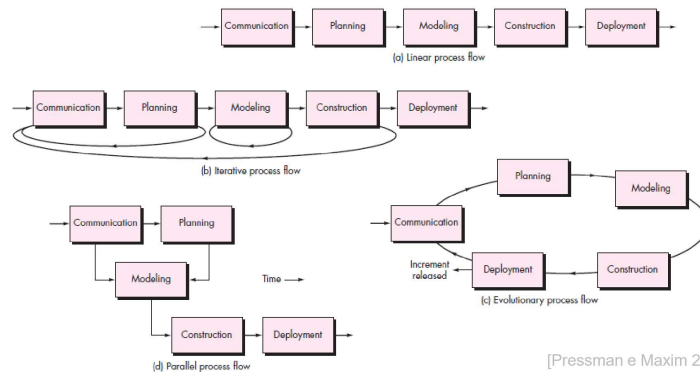
- Um processo de desenvolvimento de software tem quatro objetivos fundamentais [Booch 1994]
 - ❖ Providenciar orientação sobre a sequência de realização das atividades
 - ❖ Especificar os modelos descritivos do sistema que devem ser desenvolvidos
 - ❖ Dirigir as tarefas dos participantes e da equipa como um todo
 - ❖ Providenciar critérios para a monitorização e avaliação dos modelos e atividades do projeto

O Processo de Desenvolvimento

Características do Processo

◆ Processo

– Etapas genéricas



O Processo de Desenvolvimento

Características do Processo

- ◆ Software processes are complex and, like all intellectual and creative processes, rely on people making decisions and judgments.
- ◆ There is no ideal process and most organizations have developed their own software development processes [Sommerville 2015]
- ◆ It is pointless to look for universal notations, methods, or techniques for software engineering because different types of software require different approaches [Sommerville 2015]

O Processo de Desenvolvimento

Características do Processo

Processo de desenvolvimento de software – Será diferente de organização para organização e até mesmo de projeto para projeto, diferindo

- ❖ Nas atividades realizadas
- ❖ No número de fases
- ❖ Na ordem de execução das fases –Linear, iterativa, ...
- ❖ Nas ferramentas e métodos utilizados
- ❖ Nos produtos que precisam de ser entregues
- ❖ ...
- ❖ E em muitos outros detalhes!!!

O Processo de Desenvolvimento

Características do Processo

Sometimes, software processes are categorized as either plan-driven or agile processes [Sommerville 2015]

- ◆ Plan-driven processes are processes where all of the process activities are planned in advance and progress is measured against this plan.
- ◆ In agile processes, planning is incremental and it is easier to change the process to reflect changing customer requirements.

O Processo de Desenvolvimento

Características do Processo

- ◆ Each approach is suitable for different types of software. Generally, you need to find a balance between plan-driven and agile processes [Sommerville 2011]
- ◆ Although there is no ‘ideal’ software process, there is scope for improving the software process in many organizations [Sommerville 2015]

O Processo de Desenvolvimento

Características do Processo

- ◆ Each approach is suitable for different types of software. Generally, you need to find a balance between plan-driven and agile processes [Sommerville 2011]
- ◆ Although there is no 'ideal' software process, there is scope for improving the software process in many organizations [Sommerville 2015]

O Processo de Desenvolvimento

Modelos do processo

- ◆ Enquadramento ✓
- ◆ **Modelos não Evolutivos**
 - ❖ **Modelo em cascata (“waterfall”)**
 - ❖ Ciclo de Vida por Fases e Atividades
 - ❖ V-Model
- ◆ **Modelos Evolutivos**
 - ❖ Desenvolvimento incremental
 - ❖ Desenvolvimento em espiral
 - ❖ Unified Process
- ◆ **Métodos Ágeis**

O Processo de Desenvolvimento

Modelo primitivo

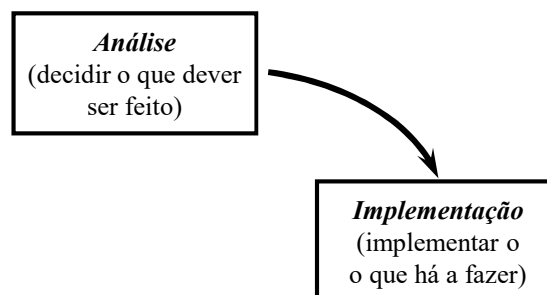
- ◆ Processo empírico não estruturado...
- ◆ Conduziu a:
 - ❖ custos elevados
 - ❖ qualidade do software muito baixa
- ◆ Insuficiente para sistemas de software complexos



O Processo de Desenvolvimento

Modelo em cascata (waterfall)

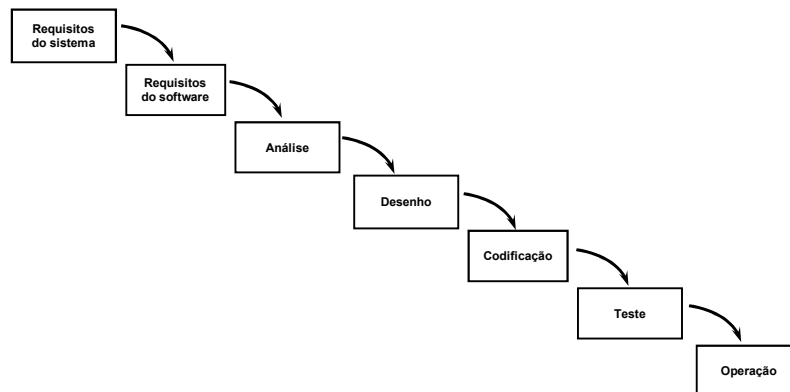
- ◆ Primeiro modelo formal adoptado na indústria
- ◆ Aborda e propõe uma sequência temporal do trabalho



O Processo de Desenvolvimento

Modelo em cascata (waterfall)

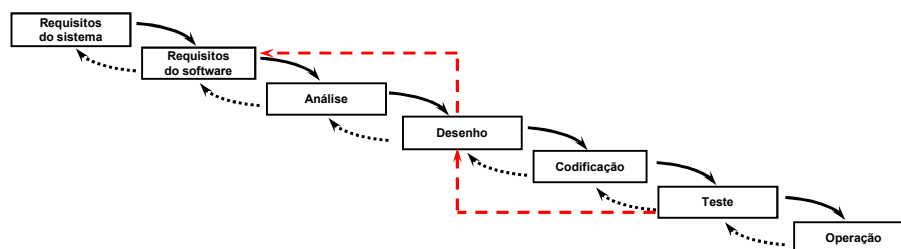
Pressupõe que cada passo é cuidadosamente executado na perfeição e não se volta a trás (como numa queda-de-água)



O Processo de Desenvolvimento

Modelo em cascata (waterfall)

- ◆ Conflito entre os passos do processo representando **progresso no tempo** e a necessidade de se **reverter** vários passos **atrás** no processo
- ◆ Reconheceu-se a necessidade de se considerarem mais dois princípios fundamentais no Processo de Desenvolvimento :
 - ❖ **fases discretas** com terminação bem definida
 - ❖ **atividades contínuas** ao longo de todo o processo



O Processo de Desenvolvimento

Modelos do processo

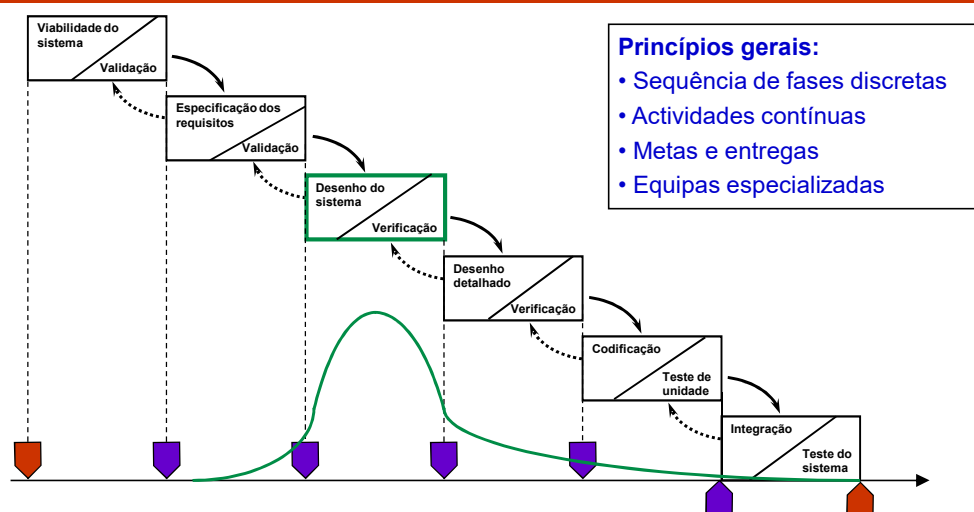
- ◆ Enquadramento ✓
- ◆ Modelos não Evolutivos ✓
 - ❖ Modelo em cascata ("waterfall") ✓
 - ❖ **Ciclo de Vida por Fases e Atividades**
 - ❖ V-Model
- ◆ Modelos Evolutivos
 - ❖ Desenvolvimento incremental
 - ❖ Desenvolvimento em espiral
 - ❖ Unified Process
- ◆ Métodos Ágeis

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 27

O Processo de Desenvolvimento

Modelo do Ciclo de Vida por Fases e Atividades



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 28

O Processo de Desenvolvimento

Modelo do Ciclo de Vida por Fases e Actividades

Características:

- ◆ *Fases discretas* – considera uma sequência rígida de fases com início e fim bem definidos. Cada fase é dedicada a um determinado tipo de trabalho (e.g. desenhar) e cobre todo o sistema de software
- ◆ *Metas (milestones) com entregas* – o final de cada fase constitui uma meta na qual é produzida uma entrega para a fase seguinte (e.g. código)
- ◆ *Actividades contínuas* – todo o tipo de trabalho é realizado ao longo do ciclo de vida, permitindo a revisão contínua de sub-produtos (entregas)
- ◆ *Matriz de fases e actividades* – especifica quais as actividades de trabalho que tomam lugar em cada fase, bem como as sub-tarefas correspondentes

O Processo de Desenvolvimento

Modelo do Ciclo de Vida por Fases e Actividades

Características:

- ◆ *Interacção entre equipas* – propõe a formação de equipas especializadas na execução das actividades, as quais interagem ao longo de todo o ciclo de vida
- ◆ *Processo contratual* – as alterações às versões iniciais das entregas são controladas sob a forma de contratos entre as equipas. Uma equipa terá de “pagar” à outra alterações que são da sua responsabilidade. Cada equipa tem um orçamento global
- ◆ *Bem estruturado* – articula de forma clara a utilização das ferramentas de desenvolvimento do software nas fases e ao longo do ciclo de vida
- ◆ *Facilita a gestão do projecto* – tem como um dos objectivos principais viabilizar a gestão do projecto, funcionando como óptima *interface* entre o desenvolvimento técnico (engenharia) e a gestão do projecto

O Processo de Desenvolvimento

Modelo do Ciclo de Vida por Fases e Actividades

Limitações:

- ◆ **Pouco flexível** – o fluxo sequencial das fases, apesar de lógico e bem estruturado, é frequentemente violado na prática, levando por vezes a que o modelo seja abandonado durante o projecto
- ◆ **Exige requisitos “estáveis”** – muitos projectos sofrem de instabilidade dos requisitos funcionais, os quais são a base de todas as fases do ciclo de vida, tornando por isso o processo de desenvolvimento instável e de difícil controlo
- ◆ **Visibilidade funcional do produto tardia** – o produto tarda a tornar-se visível, emergindo apenas nas fases finais. Isto faz com que “erros” nos requisitos só sejam descobertos nessas fases, quando é muito trabalhoso corrigi-los
- ◆ **Possíveis ineficiências** – devido à dependência entre as fases no que respeita às entregas, atrasos numa fase podem deixar “sem trabalho” as equipas das fases seguintes

O Processo de Desenvolvimento

Modelo do Ciclo de Vida por Fases e Actividades

Adequado a:

- ◆ Projectos em que os sistemas têm **requisitos funcionais estáveis** (são bem compreendidos e de forma consensual tanto pelo cliente como pela equipa de desenvolvimento)
- ◆ A equipa de desenvolvimento está **familiarizada** e tem **experiência na área de negócio** e com os **aspectos técnicos** do sistema
- ◆ Existe uma boa **relação de confiança** entre Cliente e a equipa de desenvolvimento (e.g. típico entre entidades governamentais e empresas com fraca competição)
- ◆ Em geral, o Cliente pode suportar alguns atrasos, tendo a **qualidade prioridade** sobre o **tempo** (e.g. sistema militar, hospitalar)

Modelos do processo

- ◆ Enquadramento ✓
- ◆ Modelos não Evolutivos ✓
 - ❖ Modelo em cascata (“*waterfall*”) ✓
 - ❖ Ciclo de Vida por Fases e Atividades ✓
 - ❖ **V-Model**
- ◆ Modelos Evolutivos
 - ❖ Desenvolvimento incremental
 - ❖ Desenvolvimento em espiral
 - ❖ Unified Process
- ◆ Métodos Ágeis

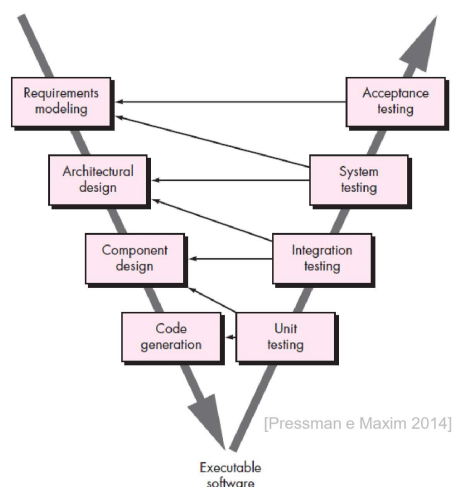
Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 33

O Processo de Desenvolvimento

V-Model

- V-model
 - Variante do waterfall model, representando as atividades de garantia da qualidade



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 34

O Processo de Desenvolvimento

Modelos do processo

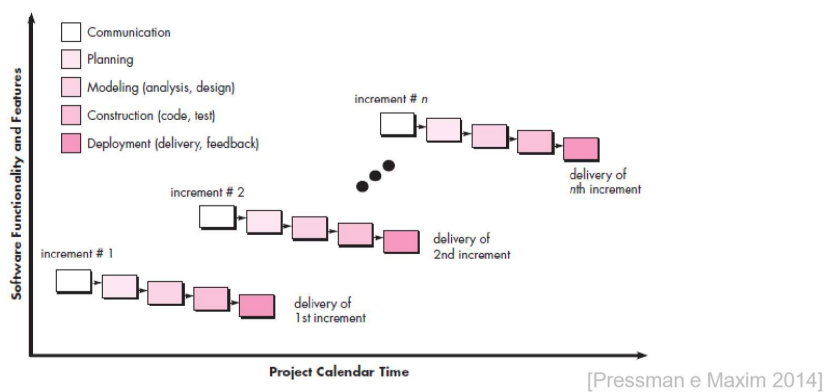
- ◆ Enquadramento ✓
- ◆ Modelos não Evolutivos ✓
 - ❖ Modelo em cascata ("waterfall") ✓
 - ❖ Ciclo de Vida por Fases e Atividades ✓
 - ❖ V-Model ✓
- ◆ Modelos Evolutivos
 - ❖ Desenvolvimento incremental
 - ❖ Desenvolvimento em espiral
 - ❖ Unified Process
- ◆ Métodos Ágeis

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 35

O Processo de Desenvolvimento

Desenvolvimento incremental



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 36

O Processo de Desenvolvimento

Desenvolvimento incremental

Problema:

- ◆ Projectos de alto risco em que a equipa de desenvolvimento não está familiarizada com a tecnologia e tipo de sistema
- ◆ Projectos que desenvolvem sistemas complexos de grande escala, onde a prototipagem é difícil ou mesmo inviável
- ◆ O Cliente necessita de alguma funcionalidade antes de ser viável desenvolver todo o sistema

Objectivos:

- ◆ Permitir entregas parciais sucessivas do sistema
- ◆ Adiar o desenvolvimento de áreas instáveis até que estas sejam clarificadas
- ◆ Acelerar o desenvolvimento do sistema
- ◆ Ganhar-se experiência antes de se desenvolverem áreas mais instáveis e críticas do sistema

O Processo de Desenvolvimento

Desenvolvimento incremental

Características:

- ◆ O desenvolvimento do sistema é dividido em vários incrementos
- ◆ Cada incremento cobre determinadas partes funcionais do sistema
- ◆ O primeiro incremento desenvolve a plataforma funcional de todo o sistema
- ◆ Cada incremento segue o ciclo de vida clássico

Estratégia:

- ◆ Desenvolvimento do sistema em incrementos de crescente funcionalidade, através de uma série de desenvolvimentos que se podem (ou não) sobrepor e no final dos quais novas versões do sistema são entregues ao Cliente. Utiliza-se frequentemente a prototipagem no início de alguns incrementos.

O Processo de Desenvolvimento

Desenvolvimento incremental

Vantagens:

- ◆ a experiência adquirida pela equipa de desenvolvimento em cada incremento é transferida para o incremento seguinte, aumentando a produtividade e qualidade do trabalho
- ◆ cada incremento beneficia do teste do incremento anterior (técnico e na sua aplicação real), bem como da experiência adquirida pelo Cliente-utilizador. Os requisitos do incremento são clarificados e o seu teste é menos complexo
- ◆ facilita a clarificação e estabilização dos requisitos ao longo de todo o projecto, bem como o acomodar de modificações
- ◆ tende a facilitar a gestão de recursos humanos ao longo de todo o projecto, devido à sobreposição dos incrementos

O Processo de Desenvolvimento

Desenvolvimento incremental

Factores críticos:

- ◆ *distribuição da funcionalidade do sistema pelos incrementos* – deve responder não só às necessidades do Cliente, mas também aos princípios básicos de uma boa arquitectura do sistema. Nem sempre é fácil de conciliar estas duas necessidades
- ◆ *mudança do Cliente* – em projectos de longa duração, a entidade financiadora do projecto (em particular entidades governamentais), pode mudar e com ela a sua perspectiva estratégica sobre a funcionalidade do sistema. Por vezes, estas mudanças radicais põem em risco uma boa arquitectura do sistema e a rentabilidade de esforços em incrementos anteriores (já existe software desenvolvido e em funcionamento que pode ter de ser refeito ou eliminado)

O Processo de Desenvolvimento

Modelos do processo

- ◆ Enquadramento ✓
- ◆ Modelos não Evolutivos ✓
 - ❖ Modelo em cascata ("waterfall") ✓
 - ❖ Ciclo de Vida por Fases e Atividades ✓
 - ❖ V-Model ✓
- ◆ Modelos Evolutivos ✓
 - ❖ Desenvolvimento incremental ✓
 - ❖ **Desenvolvimento em espiral**
 - ❖ Unified Process
- ◆ Métodos Ágeis

O Processo de Desenvolvimento

Modelo em espiral (Boehm 1988)

Problema:

- ◆ Os requisitos funcionais do sistema no seu todo são mal conhecidos
- ◆ Existem riscos diversos relacionados com aspectos tecnológicos
- ◆ O Cliente deseja uma entrega progressiva e contínua do sistema ao longo da qual possa ir definindo os requisitos
- ◆ O Cliente deseja ver o sistema a começar a funcionar o mais rapidamente possível

O Processo de Desenvolvimento

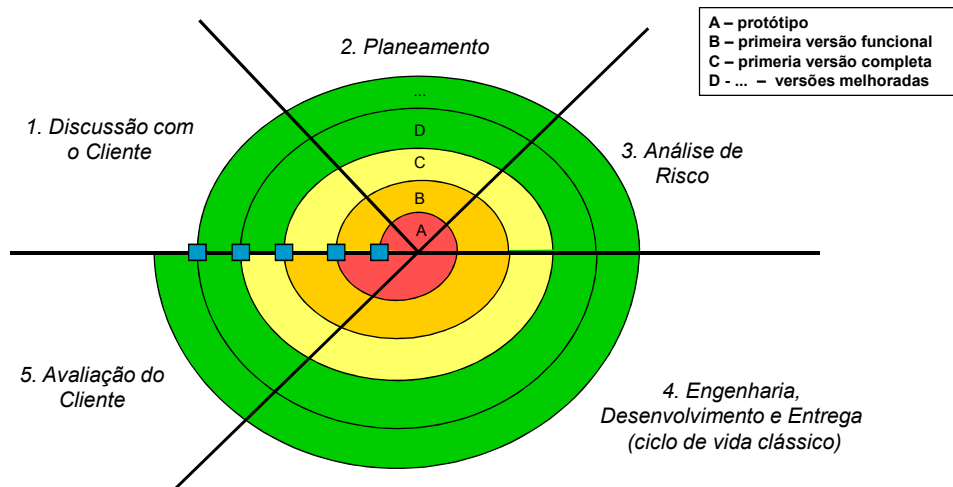
Modelo em espiral (Boehm 1988)

Objectivo: obter um desenvolvimento ...

- **Rápido** – através de versões incrementais do sistema
- **Permanente** – ao longo da vida útil do sistema
- **Controlado** – com progresso visível e medido
- **Flexível** – adaptado continuamente às necessidades do Cliente e outras mudanças
- **De baixo risco** – o conceito do produto é validado e os riscos são analisados antes do desenvolvimento das funcionalidades
- *Pretende manter a disciplina do ciclo de vida clássico, mas ultrapassando as suas desvantagens (i.e. rigidez e desenvolvimento único de uma só vez)*

O Processo de Desenvolvimento

Modelo em espiral (Boehm 1988)



O Processo de Desenvolvimento

Modelo em espiral (Boehm 1988)

- ◆ Acaba por ser equivalente a um desenvolvimento incremental contínuo, com prototipagem no primeiro incremento, mas mais sistematizado e flexível.
- ◆ Difere do desenvolvimento incremental em alguns aspectos:
 - a funcionalidade completa de todo o sistema não é especificada de início do processo
 - no início de cada ciclo incremental, na fase de discussão com o cliente, é muitas vezes utilizada a prototipagem de forma sistemática
 - potencialmente, cobre toda a vida útil do sistema
 - considera explicitamente o envolvimento do Cliente ao longo de todo o processo, no início e no final de cada ciclo
 - considera explicitamente a análise de risco antes do desenvolvimento técnico de cada nova funcionalidade
 - os incrementos são obrigatoriamente sequenciais, pelo que não se sobrepõem

O Processo de Desenvolvimento

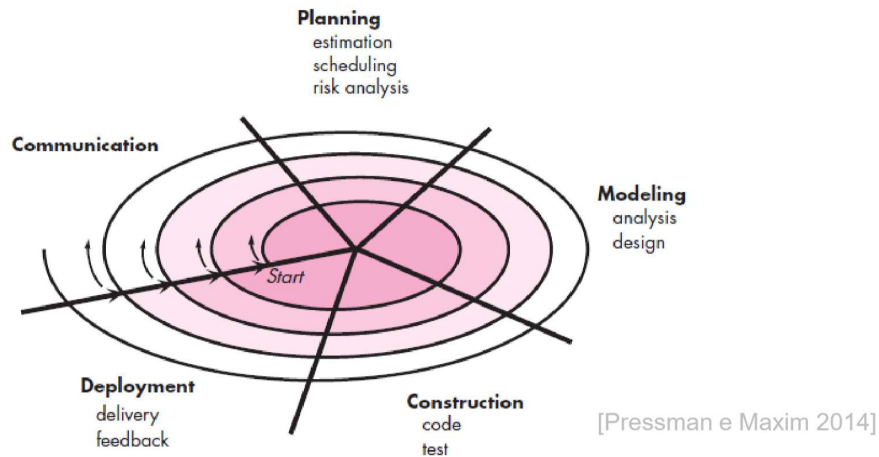
Modelo em espiral (Boehm 1988)

Possíveis dificuldades:

- ◆ o Cliente pode duvidar de que o processo evolutivo, no seu todo, seja de facto controlável (progresso, custos, recursos,...)
- ◆ exige “expertise” para a atividade de análise de risco
- ◆ se riscos importantes forem omitidos o processo pode “descarrilar”, pois cada ciclo incrementais herda os problemas dos anteriores que não tenham sido resolvidos ou descobertos.
- ◆ também relacionado com o risco existe o problema de se manter uma boa arquitetura do sistema
- ◆ *Existem muitas variações deste modelo, desenvolvidas e adotadas pelas por empresas de TIs.*

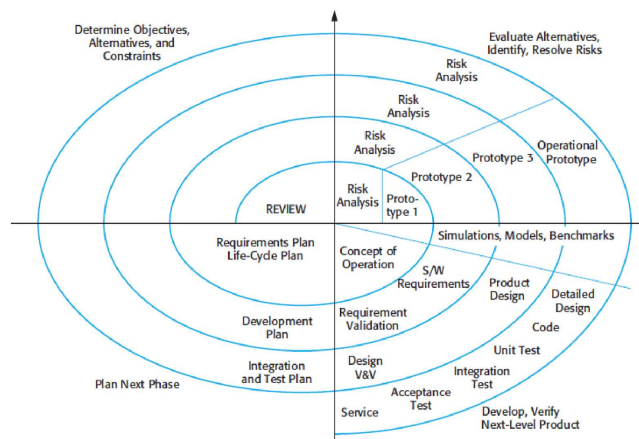
O Processo de Desenvolvimento

Modelo em espiral (Boehm 1988)



O Processo de Desenvolvimento

Modelo em espiral (Boehm 1988)



O Processo de Desenvolvimento

Modelos do processo

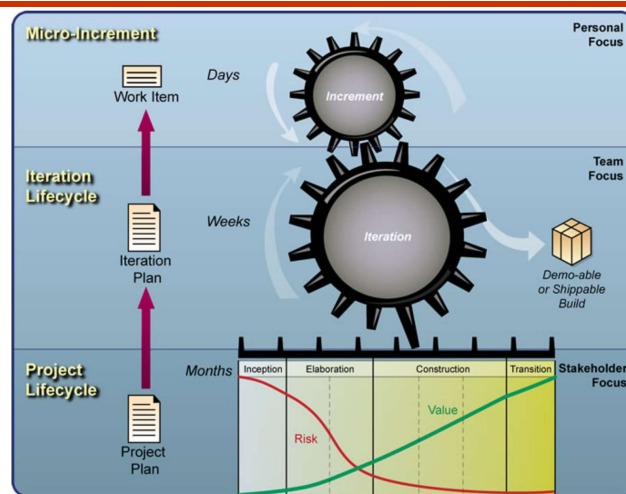
- ◆ Enquadramento ✓
- ◆ Modelos não Evolutivos ✓
 - ❖ Modelo em cascata ("waterfall") ✓
 - ❖ Ciclo de Vida por Fases e Atividades ✓
 - ❖ V-Model ✓
- ◆ Modelos Evolutivos ✓
 - ❖ Desenvolvimento incremental ✓
 - ❖ Desenvolvimento em espiral ✓
 - ❖ Unified Process
- ◆ Métodos Ágeis

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 49

O Processo de Desenvolvimento

Open Unified Process (OpenUP)



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

<http://epf.eclipse.org/wikis/openup/>

Página 50

O Processo de Desenvolvimento

Unified Process

- ◆ Usa a UML (Unified Modelling Language) como linguagem de representação.
- ◆ O Unified Process tem as seguintes características:
 - ❖ Assenta num desenvolvimento em espiral
 - ❖ Herda a perspectiva de fases e actividades do ciclo de vida clássico, propondo quatro fases ao longo das quais várias actividades tomam lugar:
 - ❖ *Inception* – concepção do produto como uma “ideia”
 - ❖ *Elaboration* – concepção da arquitectura do sistema
 - ❖ *Construction* – produção de versões “beta” do sistema
 - ❖ *Transition* – produção incremental de versões finais
- ◆ Modelos mais conhecidos RUP (Rational Unified Process) e Open Unified Process (OpenUP)

O Processo de Desenvolvimento

Unified Process

- ◆ É o modelo específico para o Processo de Desenvolvimento proposto pela originalmente pela empresa Rational Rose.
- ◆ Assenta nas seguintes práticas:
 - ❖ Desenvolvimento incremental
 - ❖ Processo centrado na Gestão de Requisitos
 - ❖ Arquitectura do software baseada em componentes
 - ❖ Modelação visual do sistema de software
 - ❖ Verificação da qualidade do software
 - ❖ Controlo de modificações
- ◆ Propõe um processo por fases discretas e actividades contínuas, herdado do Ciclo de Vida Clássico

O Processo de Desenvolvimento

Unified Process

◆ **Inception:**

- ❖ Estabelece um visão inicial para o projecto, propondo objectivos, uma definição inicial de requisitos (20%) e um plano geral para o processo (incrementos)
- ❖ Desenvolve um protótipo inicial do sistema
- ❖ Justifica o projecto na perspectiva do negócio

◆ **Elaboration:**

- ❖ Propõe uma arquitectura para o sistema com vista a eliminar riscos potenciais
- ❖ Detalha os requisitos (80%) e o plano do projecto
- ❖ Desenvolve uma versão inicial executável do sistema (protótipo evolutivo) que implementa a arquitectura de base

O Processo de Desenvolvimento

Unified Process

◆ **Construction:**

- ❖ As várias componentes funcionais são construídas em cima da arquitectura de base
- ❖ O processo de construção segue vários incrementos mais ou menos sobrepostos
- ❖ É produzida a versão "beta" pronta a instalar no cliente

◆ **Transition:**

- ❖ O sistema é instalado no cliente, a sua funcionalidade é revista e são feitas as alterações / correcções necessárias
- ❖ São feitas as conversões necessárias a partir do sistema antigo e o cliente recebe treino

O Processo de Desenvolvimento

Unified Process

- ◆ As principais vantagens do Unified Process são
 - ❖ base em princípios “seguros” de engenharia de software, tais como, abordagem iterativa, conduzida pelos requisitos e baseada na arquitetura;
 - ❖ fornece vários mecanismos, por exemplo, o protótipo no fim de cada iteração e o ponto de decisão sobre prosseguir ou não no fim de cada fase que dão visibilidade à gestão sobre o processo de desenvolvimento;
 - ❖ Conjunto bem definido de papéis e artefactos

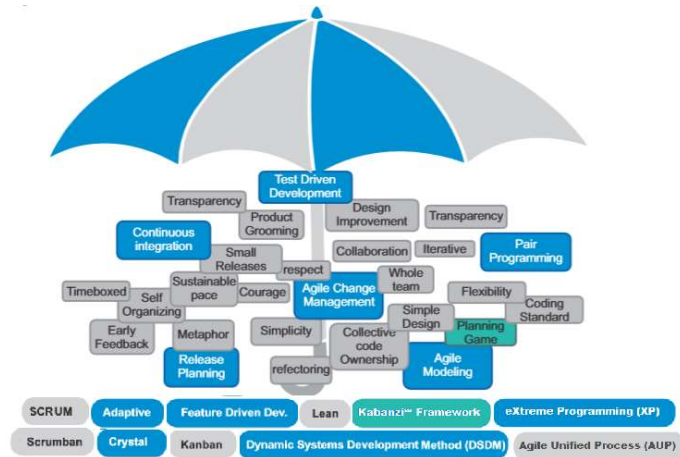
O Processo de Desenvolvimento

Modelos do processo

- ◆ Enquadramento ✓
- ◆ Modelos não Evolutivos ✓
 - ❖ Modelo em cascata (“waterfall”) ✓
 - ❖ Ciclo de Vida por Fases e Atividades ✓
 - ❖ V-Model ✓
- ◆ Modelos Evolutivos ✓
 - ❖ Desenvolvimento incremental ✓
 - ❖ Desenvolvimento em espiral ✓
 - ❖ Unified Process ✓
- ◆ **Métodos Ágeis**

Métodos Ágeis

Agile Umbrella



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 57

Métodos Ágeis

Plano

- ◆ Waterfall
- ◆ Agile
 - ❖ Scrum
 - ❖ Kanban
 - ❖ XP
 - ❖ Outros Métodos
- ◆ Conclusões

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 58

Métodos Ágeis

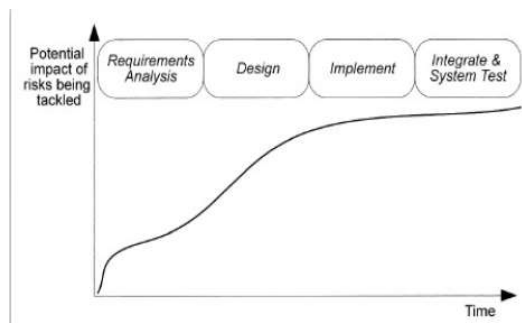
Plano

- **Waterfall**
- ◆ **Agile**
 - ❖ Scrum
 - ❖ Kanban
 - ❖ XP
 - ❖ Outros Métodos
- ◆ **Conclusões**

Métodos Ágeis

Waterfall

- Processo definido e previsível
- Fases sequenciais
- Planeamento Integral no início do projeto
- Os Riscos e as Dificuldades são agravadas com o aproximar do fim do projeto



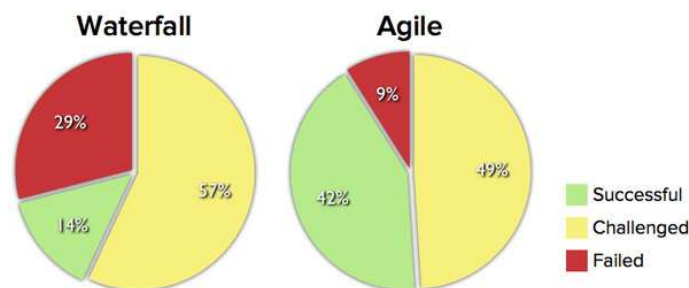
Métodos Ágeis

Waterfall, possíveis dificuldades:

- Exige requisitos estáveis
- A especificação é normalmente ambígua e pode ser interpretada de diferentes formas
- Testes realizados muito tarde
- Integração no fim do desenvolvimento
- O progresso é medido pela % de tarefas executadas

Métodos Ágeis

Waterfall vs Agile



Source: The CHAOS Manifesto, The Standish Group, 2012.

Métodos Ágeis

Outro estudo:

- Productivity and cost are impacted positively by about 10 to 20% (trended over a ten year period and compared against domain specific benchmarks);
- Agile quality does not measure up to that generated on traditional projects;
- As another example, the big three issues in agile according to the data are scaling, agile maintenance and contracting/subcontracting for agile.

(Don Reifer, 2012)

Métodos Ágeis

Plano

✓ Waterfall

➤ Agile

❖ Scrum

❖ Kanban

❖ XP

❖ Outros Métodos

◆ Conclusões

Métodos Ágeis

Agile Manifest:

- ◆ Individuals and interactions over processes and tools
- ◆ Working software over comprehensive documentation
- ◆ Customer collaboration over contract negotiation
- ◆ Responding to change over following a plan

Métodos Ágeis

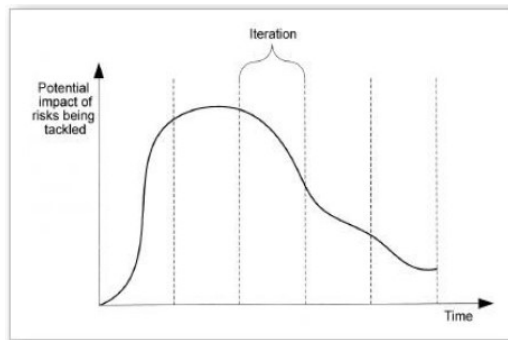
Agile:

- Aborda os projetos de forma incremental e iterativa
- Facilita mudanças no projeto
- Com frequentes entregas, aumenta o “Business Value”
- Aumenta a visibilidade do projeto para todos os intervenientes
- Progresso medido por testes efetuados com sucesso
- Melhoramento progressivo devido à observação periódica

Métodos Ágeis

Agile:

- Processos empíricos e adaptativos
- Ciclos curtos e repetitivos
- Planeamento a curto prazo baseado no feedback, inspeção e adaptação
- Desenvolvimento iterativo, diminuindo a complexidade nas fases finais do projeto



Métodos Ágeis

Princípios Ágeis (Visão Global):

- Participação de todos os intervenientes nas decisões (responsabilização)
- Todas as decisões são públicas (envolvimento)
- Mantêm terminologias comuns entre todos os intervenientes do projeto (transparência)



<http://www.criacionismo.com.br/2012/04/transparencia-nova-mentira.html>

Métodos Ágeis

Princípios Ágeis (Processo Orientado ao Cliente):

- Trabalhar na perspetiva do cliente
- Deixar o cliente/utilizador decidir o que é mais importante
- Prioridade máxima às tarefas que acrescentam **Valor ao Negócio**
- Aceitar com naturalidade a mudança



<http://orientaon.com.br/>

Métodos Ágeis

Princípios Ágeis (Colaboração Diária):

- Clientes e Equipa devem comunicar diariamente
- A comunicação deve ser, preferencialmente presencial
- Os temas devem centrar-se na:
 - Troca de informação
 - Estratégia comum

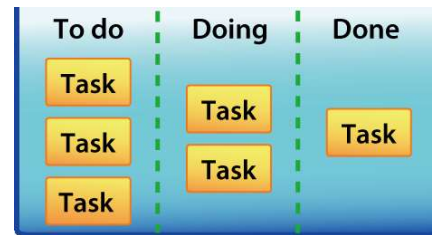


<http://nogueiramarques.com.br/>

Métodos Ágeis

Princípios Ágeis (Manter o Fluxo):

- Projeto dividido em “Timeboxes” para manter as entregas constantes
- Tarefas centralizadas numa lista, todos os membros do projeto sabem quais são as próximas tarefas
- Trabalho constante e regular mantém a tranquilidade (diminui as horas extras)



Métodos Ágeis

Princípios Ágeis (Pensar em grande, começar pelo pequeno):

- Entregar “coisas” pequenas rapidamente;
- Recolher feedback, o mais cedo possível;
- Realizar entregas com valor de forma regular.



Métodos Ágeis

Princípios Ágeis (Outros):

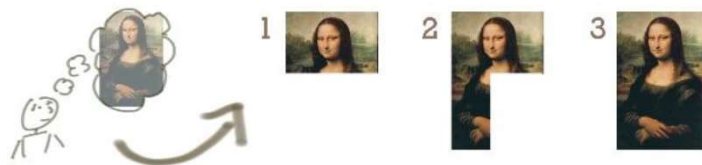
- Reflexões constantes e periódicas (Melhoria contínua)
- Eliminar os desperdícios (funcionalidades desnecessárias, requisitos não usados, ...)
- Equipas pequenas e multi-disciplinares
- Requisitos e Arquitetura evoluem ao longo do tempo
- Desenvolvimento iterativo e incremental



Métodos Ágeis

Princípios Ágeis:

Iterativo



Incremental



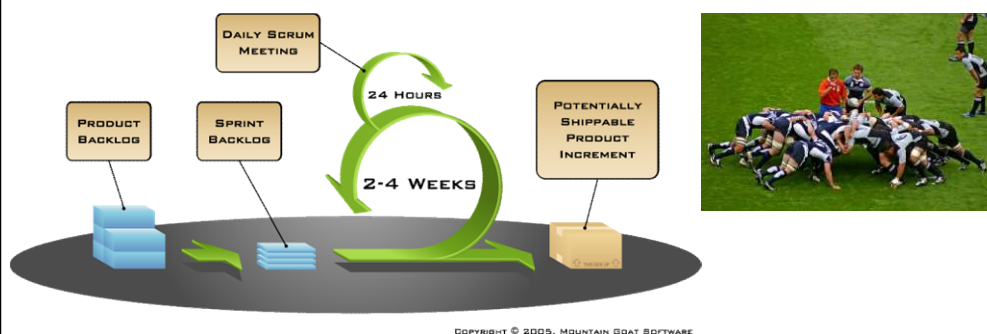
Métodos Ágeis

Plano

- ✓ Waterfall
- ✓ Agile
 - **Scrum**
 - ❖ Kanban
 - ❖ XP
 - ❖ Outros Métodos
- ◆ Conclusões

Métodos Ágeis

SCRUM:



Métodos Ágeis

Scrum Theory

- Scrum is founded on empirical process control theory, or empiricism
- Empiricism asserts that knowledge comes from experience *and* making decisions based on what is known.

Three pillars of Scrum:

- Transparency
- Inspection
- Adaptation

Métodos Ágeis – Artefactos do Scrum

User Story

- ◆ Representam um objetivo a ser alcançado (feature);

Seguindo Mike Cohn...

Exemplo:

“As a User I want to have access to detailed billing information so that I can manage the month billing value”

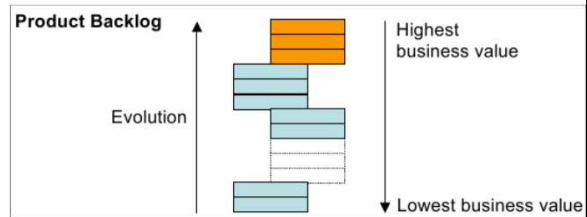
As a <User Role>
I want to <Action or Goal>
So that <Business Value>

www.santiaal.com/java

Métodos Ágeis – Artefactos do Scrum

Product Backlog

- Lista de User Stories
- Ordenada Por “Business Value”
- Stories mais detalhadas no início da lista
- Priorizada pelo “Product Owner”



Métodos Ágeis – Artefactos do Scrum

Sprint Backlog

- The Sprint Backlog is the set of Product Backlog items selected for the Sprint, plus a plan for delivering the product Increment and realizing the Sprint Goal
- The Sprint Backlog makes visible all of the work that the Development Team identifies as necessary to meet the Sprint Goal.
- The Sprint Backlog is a highly visible, real-time picture of the work that the Development Team plans to accomplish during the Sprint.

Métodos Ágeis – Artefactos do Scrum

Increment

- The Increment is the sum of all the Product Backlog items completed during a Sprint and the value of the increments of all previous Sprints.
- At the end of a Sprint, the new Increment must be “Done,” which means it must be in useable condition and meet the Scrum Team’s definition of “Done.”

Métodos Ágeis – Sprints no Scrum

- The heart of Scrum is a Sprint, a time-box of one month or less during which a “Done”, useable, and potentially releasable product Increment is created.
- Sprints contain and consist of the Sprint Planning, Daily Scrums, the development work, the Sprint Review, and the Sprint Retrospective.
- A Sprint would be cancelled if the Sprint Goal becomes obsolete.
- Due to the short duration of Sprints, cancellation rarely makes sense.

Métodos Ágeis – Papeis no Scrum

Product Owner

- Um por projeto
- Toma as decisões do produto/projeto
- É o responsável pela BackLog de Produto
- Tem a última palavra no planeamento do Incremento / sprint
- Tem a obrigação de explicar as User Stories à equipa

Métodos Ágeis – Papeis no Scrum

Scrum Master

- Um por equipa
- Responsável por transmitir os valores do Scrum
- Garante que o processo é seguido
- Responsável por resolver os impedimentos
- É o líder da equipa (facilitador)

Métodos Ágeis – Papeis no Scrum

Scrum Team

- Equipas pequenas mais de 3 e menos de 9 (ideal 5 a 9)
- Equipas multidisciplinares (programadores, testes, BAs, etc)
- Poder de decisão, mas também mais responsabilidade
- Equipas Auto-Organizadas
- As equipas devem ser estáveis e alocadas a 100% ao projeto

Métodos Ágeis – Reuniões no Scrum

Daily Scrum

- Reunião da equipa em pé
- Reunião Diária a horas fixas
- Não deve durar + de 15 min
- Cada membro deve dizer:
 - O que fez ontem ?
 - O que tem para fazer hoje ?
 - Que impedimentos tem ?



Métodos Ágeis – Reuniões no Scrum

Sprint Planning

- Planeamento do próximo “Sprint”
- É decidido quais as User Stories que serão entregues
- Todos os membros têm de se comprometer com as User Stories
- O PO leva a backlog priorizada
- O Scrum Master atua como um facilitador



Métodos Ágeis – Reuniões no Scrum

Sprint Review

- The Development Team discusses what went well during the Sprint, what problems it ran into, and how those problems were solved;
- The Development Team demonstrates the work that it has “Done” and answers questions about the Increment;
- The Product Owner discusses the Product Backlog as it stands. He or she projects likely completion dates based on progress to date (if needed);
- The entire group collaborates on what to do next, so that the Sprint Review provides valuable input to subsequent Sprint Planning;

Métodos Ágeis – Reuniões no Scrum

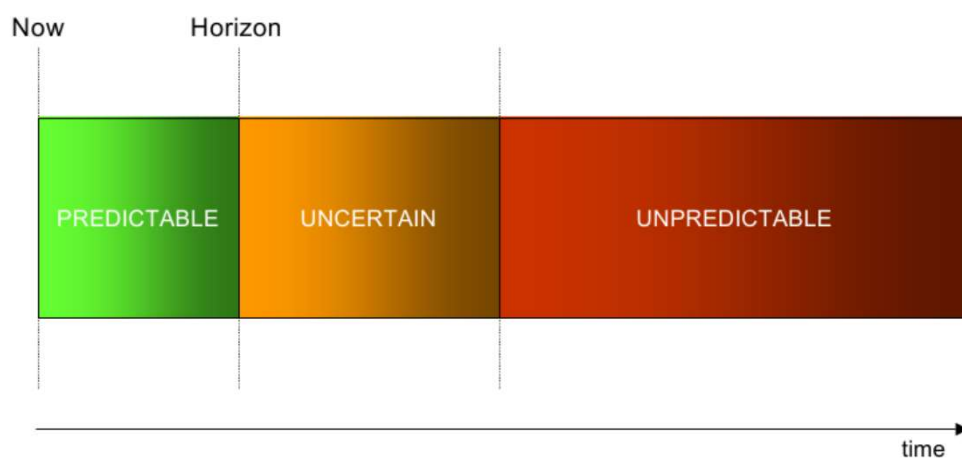
Sprint Retrospective

The Sprint Retrospective is an opportunity for the Scrum Team to inspect itself and create a plan for improvements to be enacted during the next Sprint.

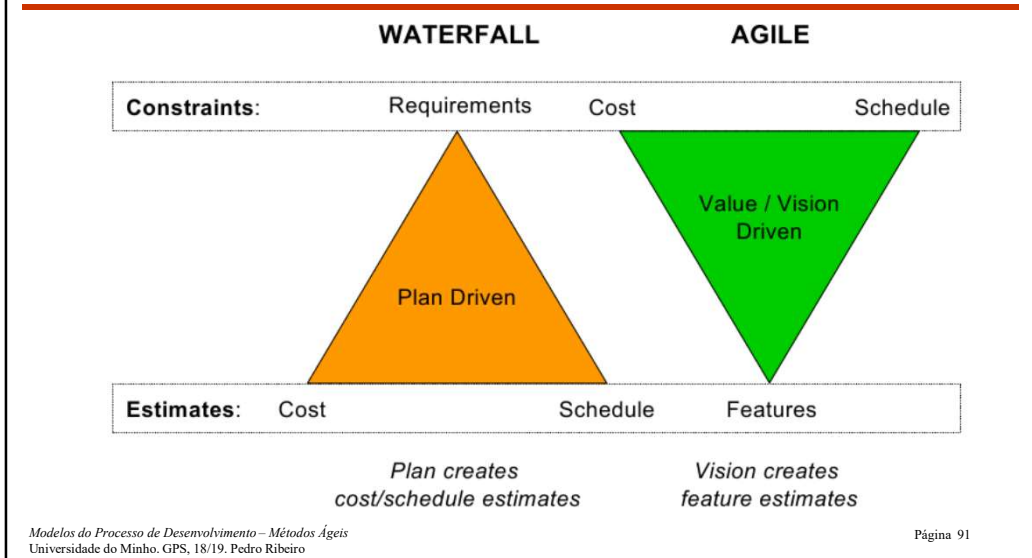
The purpose of the Sprint Retrospective is to:

- ◆ Inspect how the last Sprint went with regards to people, relationships, process, and tools;
- ◆ Identify and order the major items that went well and potential improvements; and,
- ◆ Create a plan for implementing improvements to the way the Scrum Team does its work.

Métodos Ágeis – Planeamento no Scrum



Métodos Ágeis – Planeamento no Scrum



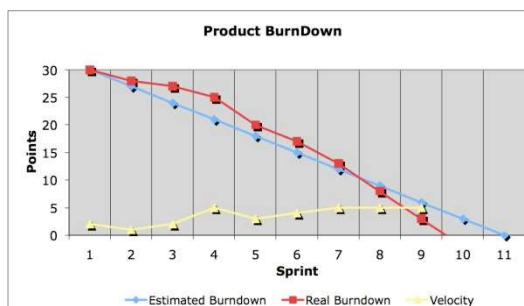
Métodos Ágeis – Planeamento no Scrum

BurnDown Charts

- Indicador de progresso do “Sprint”
- Permite à equipa ver a evolução e, caso seja necessário, tomar medidas
- Deve estar visível para todos

Vertical: Story Points

Horizontal: Dias (Sprints)



Métodos Ágeis – Planeamento no Scrum

Velocity

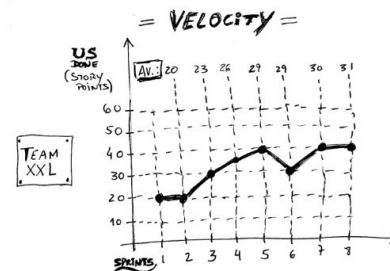
- ◆ Número de Story Points que a equipa consegue entregar num Sprint
- ◆ Usado para medir as datas em que as User Stories são entregues ou a finalização do projeto

Exemplo:

Backlog tem 500 Story Points

Por Sprint (2 semanas) são efetuados 50 SP

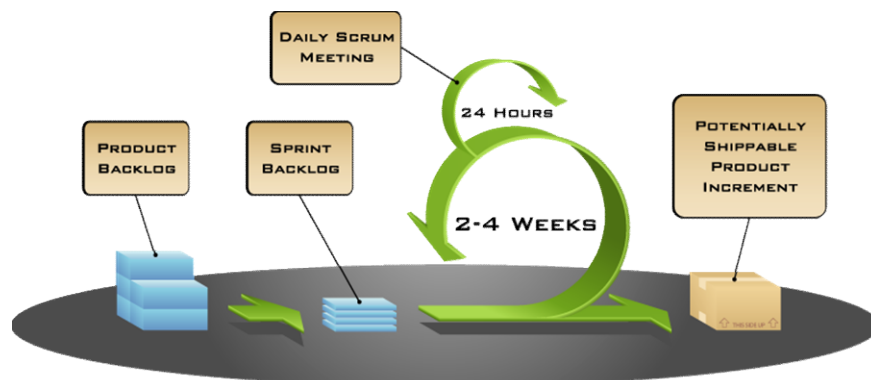
Ou seja, projeto demora $2 * 10$ Semanas



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Métodos Ágeis

SCRUM:



COPYRIGHT © 2005, MOUNTAIN GOAT SOFTWARE

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 94

Métodos Ágeis

Plano

- ✓ Waterfall
- ✓ Agile
 - ✓ Scrum
 - ❖ **Kanban**
 - ❖ XP
 - ❖ Outros Métodos
- ◆ Conclusões

Métodos Ágeis - Kanban

- ◆ Desenvolvido por Taichi Ohno na Toyota nos anos 40;
- ◆ Sistema de escalonamento para uma produção *lean* e *Just-In-Time* (JIT);
- ◆ Maximizar o valor para o cliente minimizando os desperdícios;
- ◆ Melhorar continuamente o processo de produção.

Métodos Ágeis - Kanban

Quatro passos do Kanban:

1. Visualizar o trabalho;
2. Limitar o WIP (Work In Progress);
3. Controlo do fluxo baseado no *pull*;
4. Monitorizar, adaptar e melhorar.

◆ Ferramenta principal: quadro branco com post-it's

Métodos Ágeis - Kanban

1. Visualizar o Trabalho

- ◆ Identificar o fluxo de trabalho e desenhar uma coluna para cada passo;
- ◆ Escrever cada tarefa em post-it's separados. Colocar as tarefas no quadro.



(en.wikipedia.org)

Métodos Ágeis - Kanban

1. Visualizar o Trabalho (exemplo)



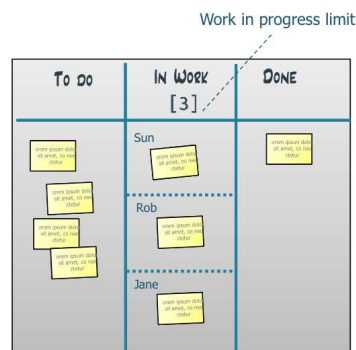
Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

(leankit.com)
Página 99

Métodos Ágeis - Kanban

2. Limitar o WIP (Work In Progress);

- ◆ Definir limites nas colunas;
- ◆ Pretende-se um fluxo elevado mas suave e eliminação de desperdícios.



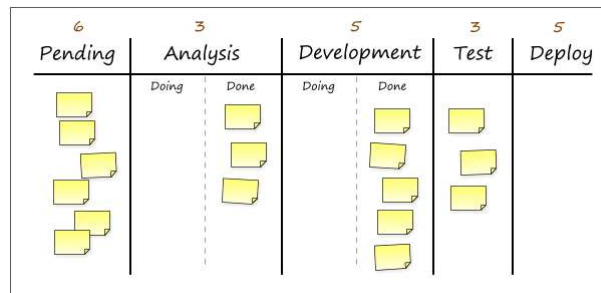
Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

(technitai.wordpress.com)
Página 100

Métodos Ágeis - Kanban

3. Controlo do fluxo baseado no pull

- ◆ As equipas têm desempenhos diferentes;
- ◆ Para resolver este problema, a equipa seguinte puxa o trabalho apenas quando o pode executar.



Modelos do Processo de Desenvolvimento – métodos ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

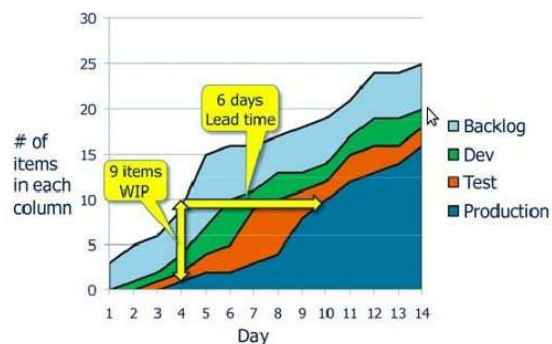
(kanbanblog.com)

Página 101

Métodos Ágeis - Kanban

4. Monitorizar, adaptar e melhorar

- ◆ Para medir o desempenho utilizar o Cumulative Flow Chart;
- ◆ Para cada dia, para cada coluna marcar as tarefas que está lá.



Modelos do Processo de Desenvolvimento – métodos ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

(csolfabok.com-)

Página 102

Métodos Ágeis - Kanban

Scrum vs Kanban

	SCRUM	KANBAN
Cadence	Regular fixed length sprints (ie, 2 weeks)	Continuous flow
Release methodology	At the end of each sprint if approved by the product owner	Continuous delivery or at the team's discretion
Roles	Product owner, scrum master, development team	No existing roles. Some teams enlist the help of an agile coach.
Key metrics	Velocity	Cycle time
Change philosophy	Teams should strive to not make changes to the sprint forecast during the sprint. Doing so compromises learnings around estimation.	Change can happen at any time

Trabalho prático (grupo: máximo 4 elementos)

Considere uma equipa que faz manutenção de sites numa empresa de TIs. A equipa é constituída por 8 elementos (1 administrativo, 1 profissional para estimativas, 4 programadores, 1 tester e 1 volante). Pretende-se utilizar o Kanban para melhorar o funcionamento da equipa. Atualmente funciona numa base FIFO.

1. Descreva os pressupostos considerados;
2. Aplique o Kanban ao processo;
3. Verifique se a equipa com o novo processo consegue satisfazer os pedidos, sabendo que:
 - ❖ A equipa recebe em média 60 pedidos por semana (duração entre 4h e 12h);
 - ❖ Em média o tester demora duas horas por pedido;
 - ❖ Cada pedido tem uma prioridade por tipo e uma prioridade por cliente

(Fazer o upload do ficheiro no Blackboard)

Métodos Ágeis

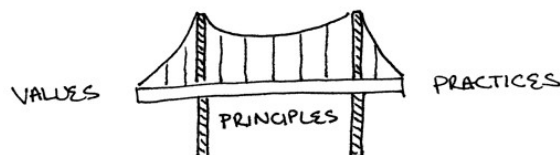
Plano

- ✓ Waterfall
- ✓ Agile
 - ✓ Scrum
 - ✓ Kanban
 - ❖ **XP**
 - ❖ Outros Métodos
- ◆ Conclusões

Métodos Ágeis – XP

Introdução ao XP

- ◆ Criado nos anos 90 por Ken Beck
- ◆ Orientado à mudança
- ◆ Ciclos curtos
- ◆ Orientado aos testes



Métodos Ágeis - XP

Valores

- ◆ Comunicação
- ◆ Feedback
- ◆ Simplicidade
- ◆ Coragem
- ◆ Respeito

Métodos Ágeis - XP

Comunicação

- ◆ Compreensão clara dos benefícios para o negócio.
Comunicação entre os *players* principais na organização cliente.
- ◆ Comunicação entre a equipa de desenvolvimento e o cliente. Renovação frequente dos requisitos por parte do cliente e informação ao cliente sobre o estado do projeto
- ◆ Comunicação entre os membros da equipa de desenvolvimento. Todos envolvidos, atualizados em relação ao progresso e com objetivos definidos.

Métodos Ágeis - XP

Feedback

- ◆ Muito ligado à comunicação
- ◆ Necessidade de mecanismos reais para manter o cliente informado e envolvido. Segurança que estamos a construir a solução certa para o negócio
- ◆ Na equipa de desenvolvimento é necessário que todos saibam o que se passa. Os elementos também precisam de saber a qualidade do seu trabalho e a qualidade do trabalho dos outros membros.

Métodos Ágeis - XP

Simplicidade

- ◆ *Features* que não interessam não devem ser desenvolvidas
- ◆ Os requisitos surpreendentes são ameaças potenciais para o sucesso do projeto. Estamos a criar complexidade extra na solução.
- ◆ É necessário ponderar bem se o tempo e custo para desenvolver a nova feature compensa. Tem mesmo valor para o negócio

Métodos Ágeis - XP

Coragem

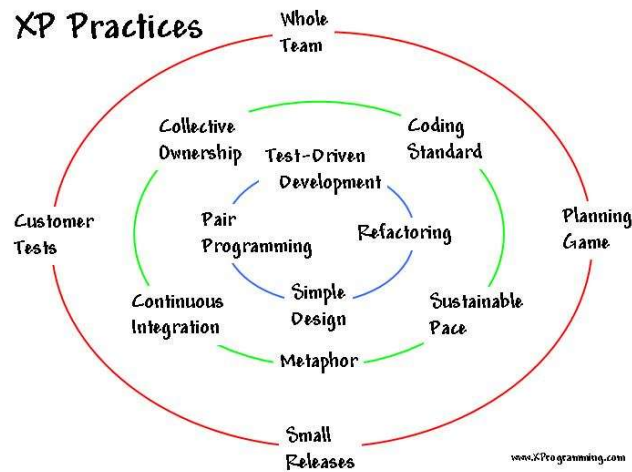
- ◆ Confiança para fazer “coisas” que noutros ambientes seriam consideradas de elevado risco
- ◆ Alinhamento com a mudança. Alterações do cliente mas também alterações ao próprio código.
- ◆ Como os desportos radicais, é o desenvolvimento de software sem as restrições habituais. Tal como nos desportos radicais é necessária uma grande disciplina.

Métodos Ágeis - XP

Respeito

- ◆ Muitos dos problemas no desenvolvimento de software relacionam-se com pessoas e normalmente com o relacionamento interpessoal.
- ◆ Para ultrapassar estes problemas é necessário tratar cada indivíduo com respeito. Naturalmente a responsabilidade e confiança têm que estar sempre presentes.

Métodos Ágeis - XP



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

www.xprogramming.com

Página 113

Métodos Ágeis - XP

1. Test-Driven Development

- ◆ Desenvolver os testes antes de codificar
- ◆ Um mecanismo rápido para ter feedback sobre a correção do código
- ◆ Se qualquer teste unitário falhar o código tem que ser corrigido
- ◆ Testes funcionais desenvolvidos com o cliente
- ◆ Os testes funcionais medem o progresso

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 114

Métodos Ágeis - XP

2. Pair Programming

- ◆ Duas pessoas, uma máquina. Todo o código deve ser produzido desta forma;
- ◆ Processo de revisão contínua e diminuição dos erros cometidos;
- ◆ Os pares mudam de forma regular;
- ◆ Excelente mecanismo de aprendizagem;
- ◆ Maior confiança no trabalho realizado;

Métodos Ágeis - XP

3. On-site Customer

- ◆ Permite clarificar rapidamente as *user-stories*;
- ◆ A equipa de desenvolvimento não tem que tomar pressupostos
- ◆ A comunicação cara a cara deve permitir diminuir os mal-entendidos;
- ◆ A equipa de desenvolvimento não tem que esperar pelas decisões do cliente

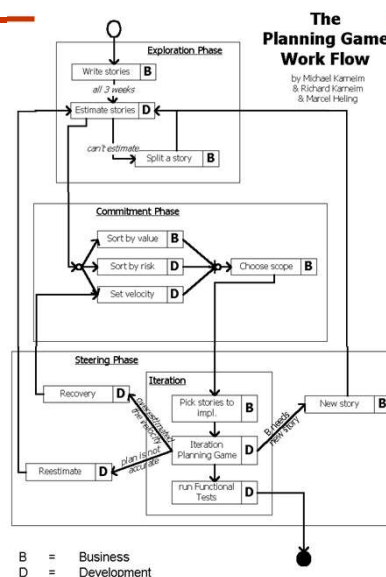
Métodos Ágeis - XP

4. The Planning Game

- ◆ Requisitos visíveis através de *user-stories* (pequenas partes de funcionalidades com significado para o cliente e equipa);
- ◆ Recursos e risco das user-stories estimados pela equipa de desenvolvimento;
- ◆ User-stories priorizadas pelo cliente;
- ◆ O Planning Game é executado antes de cada release.

Métodos Ágeis - XP

4. The Planning Game



Métodos Ágeis - XP

User story:

The screenshot shows a 'Edit User Story' form for 'US9: Credit card payments'. Annotations point to various fields:

- Story name:** Points to the 'Name' field containing 'Credit card payments'.
- Value statement:** Points to the 'Description' field containing 'As a purchaser on the website, I want the ability to pay with a credit card, so that I may immediately confirm my purchase.'
- Acceptance Criteria:** Points to the 'Acceptance Criteria' section containing a bulleted list of requirements.
- Definition of Done:** Points to the 'Definition of Done' section containing a bulleted list of completion criteria.
- Attach details and documents when necessary:** Points to the 'Attachments' section with a 'Browse...' button.
- What is required for the business and product owner to accept the story:** Points to the 'Acceptance Criteria' section.
- What is required by the team (quality/standards) before sending out for review. Does not change from one story to another. Mature teams may post this on the wall of the team working area instead of within each story.** Points to the 'Definition of Done' section.
- Size (effort) estimate, in relative points:** Points to the 'Points' field in the 'Schedule' section.

Other visible fields include 'ID: 459', 'Owner: Greg', 'Status: Defined', 'Iteration: Unscheduled', 'Plan Est: 8.0', 'To Do: 0.0 Hours', and 'Task Est: 0.0 Hours'.

Modelos do Processo de Desenvolvimento -
Universidade do Minho. GPS, 18/19. Pedro

(Ronica Roth)

Métodos Ágeis - XP

5. Metaphor

- ◆ Orienta todo o desenvolvimento e comunicação com uma alegoria simples do funcionamento global do sistema;
- ◆ Permite Partilhar a visão, a terminologia e a arquitetura;
- ◆ *“The metaphor just helps everyone on the project understand the basic elements and their relationships. Words chosen to identify technical entities should be consistently taken from the chosen metaphor. As development proceeds and the metaphor matures, the whole team will find new inspiration from examining the metaphor.”*

(Ken Beck, *Extreme Programming Explained*)

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 120

Métodos Ágeis - XP

6. Small Releases

- ◆ *Releases* o mais pequenas possível, mas mesmo assim entregando valor para o negócio;
- ◆ Realizar o *Planning Game* depois de cada iteração;
- ◆ Uma *release* não é um protótipo. É real, funcionalmente útil, implementa novas histórias e é completamente testada;
- ◆ Novas *releases* com intervalos de uma a 3 semanas

Métodos Ágeis - XP

7. Simple Design

- ◆ Conceber inicialmente “coisas” simples e a funcionar para obter *feedback* o mais cedo possível;
- ◆ Simplificar a conceção não significa “não conceber”;
- ◆ Conceber da forma mais simples possível, caso seja necessário faz-se o *refactoring* mais tarde;
- ◆ Não perder tempo a conceber aspetos que não vamos precisar no curto prazo.

Métodos Ágeis - XP

8. Continuous Integration

- ◆ O código deve ser integrado no sistema no mínimo, várias vezes por dia;
- ◆ Todos os testes unitários devem ser realizados pelo par responsável pelo código antes da integração
- ◆ A integração contínua torna-se numa fonte de confiança de que a equipa está a avançar;

Métodos Ágeis - XP

9. Coding Standards

- ◆ Definição de regras para todo o código partilhado;
- ◆ Regras também para os testes e *user stories*;
- ◆ Todas as regras devem ser partilhadas e aprovadas no início do projeto;
- ◆ Normas simples e práticas. Alicerces do refactoring e da partilha de código.

Métodos Ágeis - XP

10. Collective Code Ownership

- ◆ Todo o código pertence a todos os programadores;
- ◆ Qualquer elemento pode alterar qualquer linha de código;
- ◆ Espera-se código de melhor qualidade (qualquer um pode melhorar qualquer parte) e um progresso mais rápido.

Métodos Ágeis - XP

11. Refactoring

- ◆ Consiste em reestruturar o código sem alterar a sua funcionalidade;
- ◆ O *refactoring* do código pode obrigar ao *refactoring* dos testes unitários correspondentes;
- ◆ O *refactoring* está ligado principalmente à simplificação do código e melhoria do design.

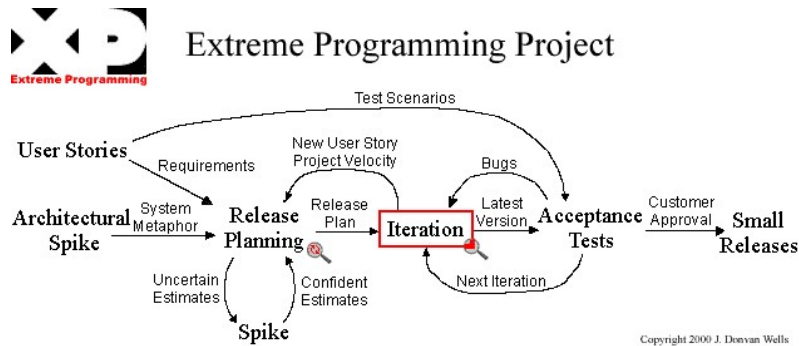
Métodos Ágeis - XP

12. Forty-Hour Week

- ◆ Programadores cansados escrevem código fraco e com mais erros;
- ◆ Espera-se que haja uma diminuição significativa do stress típico, com melhor comunicação, feedback constante, maior qualidade e outras práticas;
- ◆ O planeamento mais realista e com menor incerteza (curto prazo) ajuda muito.

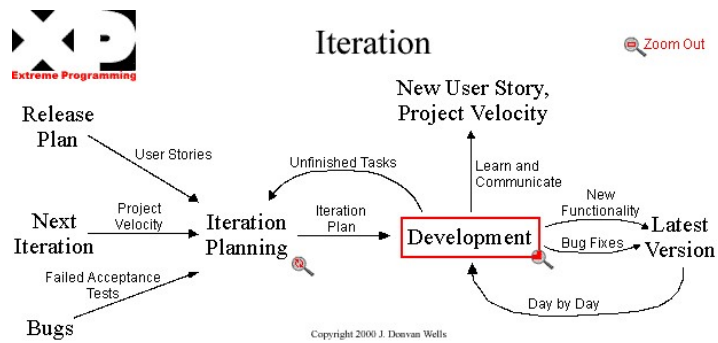
Métodos Ágeis – XP

Modelo XP



Métodos Ágeis – XP

Iterações no XP

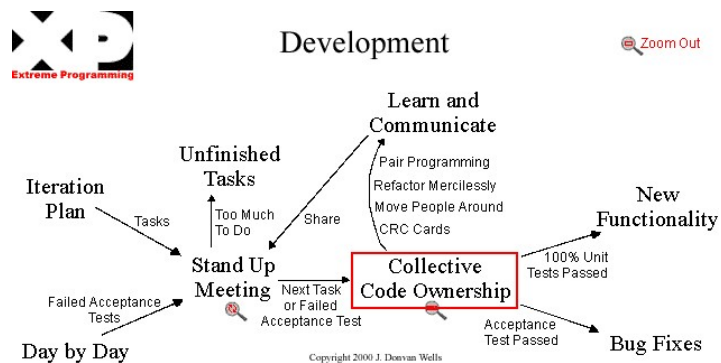


Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

(www.extremprogramming.org)
Página 129

Métodos Ágeis – XP

Foco na comunicação:

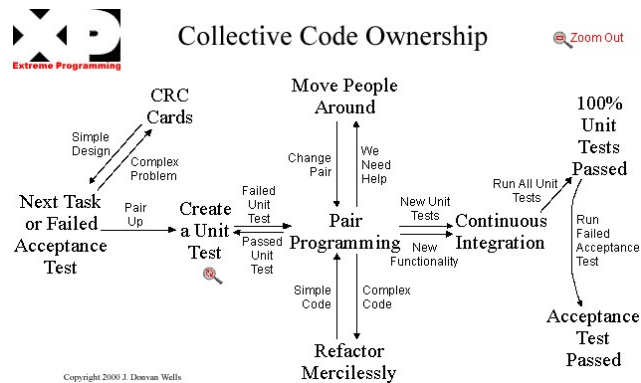


Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

(www.extremprogramming.org)
Página 130

Métodos Ágeis – XP

Desenvolvimento Orientado ao Teste (TDD):



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

(www.extremprogramming.org)
Página 131

Métodos Ágeis

Plano

- ✓ Waterfall
- ✓ Agile
 - ✓ Scrum
 - ✓ Kanban
 - ✓ XP
- ❖ **Outros Métodos**
- ◆ **Conclusões**

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 132

Métodos Ágeis – Outros métodos

Features Driven Development (FDD)

Combina a entrega iterativa da solução com práticas de desenvolvimento de software, através de:

- ❖ Equipas a modelar a necessidade de negócio antecipadamente;
- ❖ Decomposição do modelo em features menores e conjuntos de features;
- ❖ Releases iterativas com novos conjuntos de features integradas sucessivamente;
- ❖ Foco na colaboração com o cliente, produção de releases tangíveis e gestão da qualidade.

Métodos Ágeis – Outros métodos

Lean Development

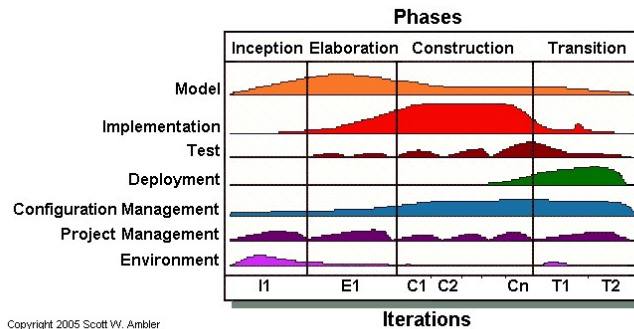
Alinhado com os princípios KAIZEN e o TQM (Total Quality Management):

- ❖ Usa a cadeia de valor para entregar as features com maior benefício para o negócio;
- ❖ Utiliza práticas Just-In-Time(JIT) para entregar os resultados o mais rápido possível;
- ❖ Gestão da qualidade rigorosa através da verificação de integridade e melhoria contínua;
- ❖ Capacita equipas competentes e interdepartamentais para entregar os resultados do máximo valor.

Métodos Ágeis – Outros métodos

AUP (Agile Unified Process)

Baseado no Unified Process mas com uma estrutura simplificada:



Métodos Ágeis – Outros métodos

Outras sugestões:

- ◆ Adaptive software development (ASD)
- ◆ Crystal Clear Methods
- ◆ Disciplined agile delivery (DAD)
- ◆ Dynamic systems development method (DSDM)
- ◆ Extreme programming (XP)
- ◆ Scrum ban
- ◆ EssUP (Essential Unified Process)

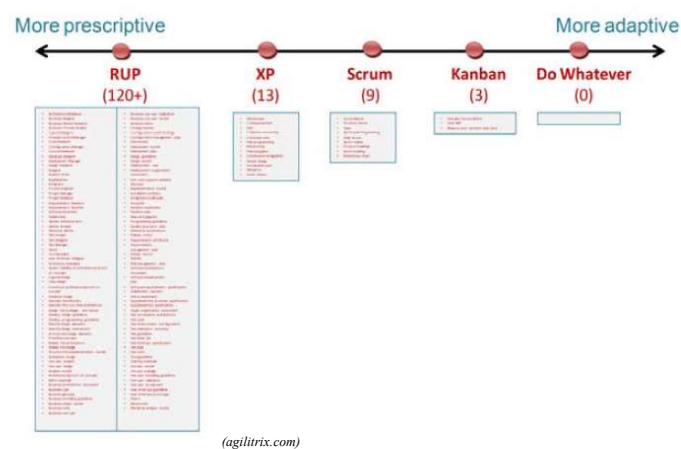
Métodos Ágeis

Plano

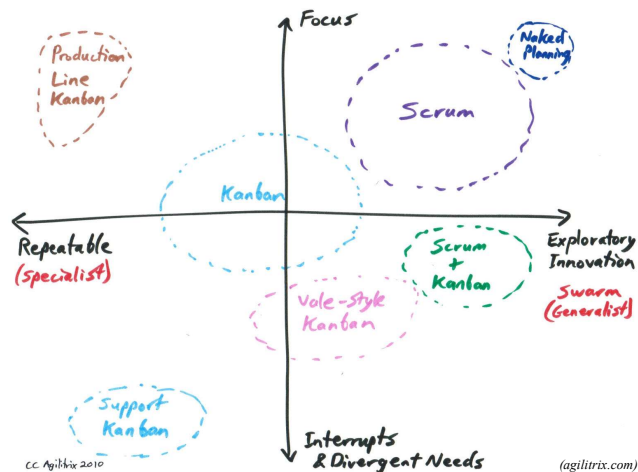
- ✓ Waterfall
- ✓ Agile
 - ✓ Scrum
 - ✓ Kanban
 - ✓ XP
 - ✓ Outros Métodos

◆ Conclusões

Métodos Ágeis - Conclusões



Métodos Ágeis - Conclusões

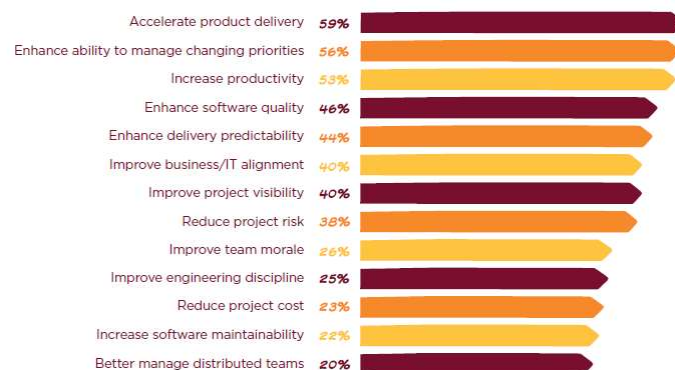


Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 139

Métodos Ágeis - Conclusões

Razões para adotar os ágeis:

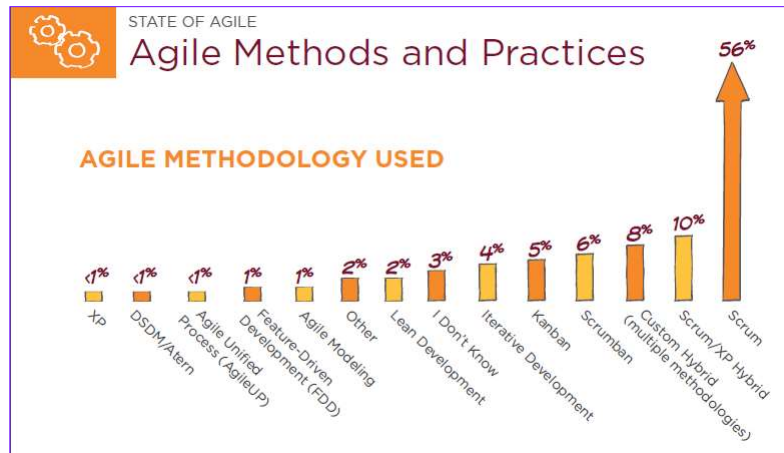


Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

(versionone.com)

Página 140

Métodos Ágeis - Conclusões



Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

(versionone.com)
Página 141

Métodos Ágeis - Conclusões

O que falta ao Ágil?

- ◆ Não tem gestão de subcontratações;
- ◆ Não tem gestão de configurações;
- ◆ Não tem gestão de risco.
- ◆ ...

Mas os ágeis não dizem que não se pode fazer... se for necessário de ser feito de forma adequada.

Modelos do Processo de Desenvolvimento – Métodos Ágeis
Universidade do Minho. GPS, 18/19. Pedro Ribeiro

Página 142

Métodos Ágeis - Conclusões

Resumo

- ◆ Mínimo de documentação;
- ◆ Foco no trabalho em equipa;
- ◆ Processo simples e empírico;
- ◆ Foco no benefício para o negócio;
- ◆ Entregas regulares com ciclos curtos;
- ◆ Orientado à mudança.

Métodos Ágeis

Referências:

- ❖ Agile Manifesto: www.agilemanifesto.org
- ❖ Agile Alliance: www.agilealliance.com
- ❖ Agile Journal: www.agilejournal.com
- ❖ AgileKiwi: www.agilekiwi.com
- ❖ Agilitrix: <http://agilitrix.com/2010/05/scrum-or-kanban-yes/>
- ❖ Agile Software Development: www.agilesoftwaredevelopment.com
- ❖ Alistair Cockburn: <http://alistair.cockburn.us/>
- ❖ Scrum: <http://www.scrumalliance.org/>
- ❖ FDD: www.influxive.com/fdd-overview.html
- ❖ Lean: www.leanprimer.com/downloads/lean_primer.pdf
- ❖ XP: www.extremeprogramming.org
- ❖ Kanban: www.infoq.com/minibooks/kanbanscrum-minibook

Gestão do Processo de Software

Modelos do Processo de Desenvolvimento (Modelos Ágeis)

FIM

Pedro Abreu Ribeiro

pmgar@dsi.uminho.pt

Departamento de Sistemas de Informação
Universidade do Minho