

Desenvolvimento ágil de Software

Joaquim Lopes Júnior

joaquim@f6sistemas.com.br

Diretor PHPrime Training e F6 Sistemas

Bacharel em C. Da Computação – UFMG – 2006

Mestrado (em andamento) – UFMG – 2009



1º Encontro PHPMG

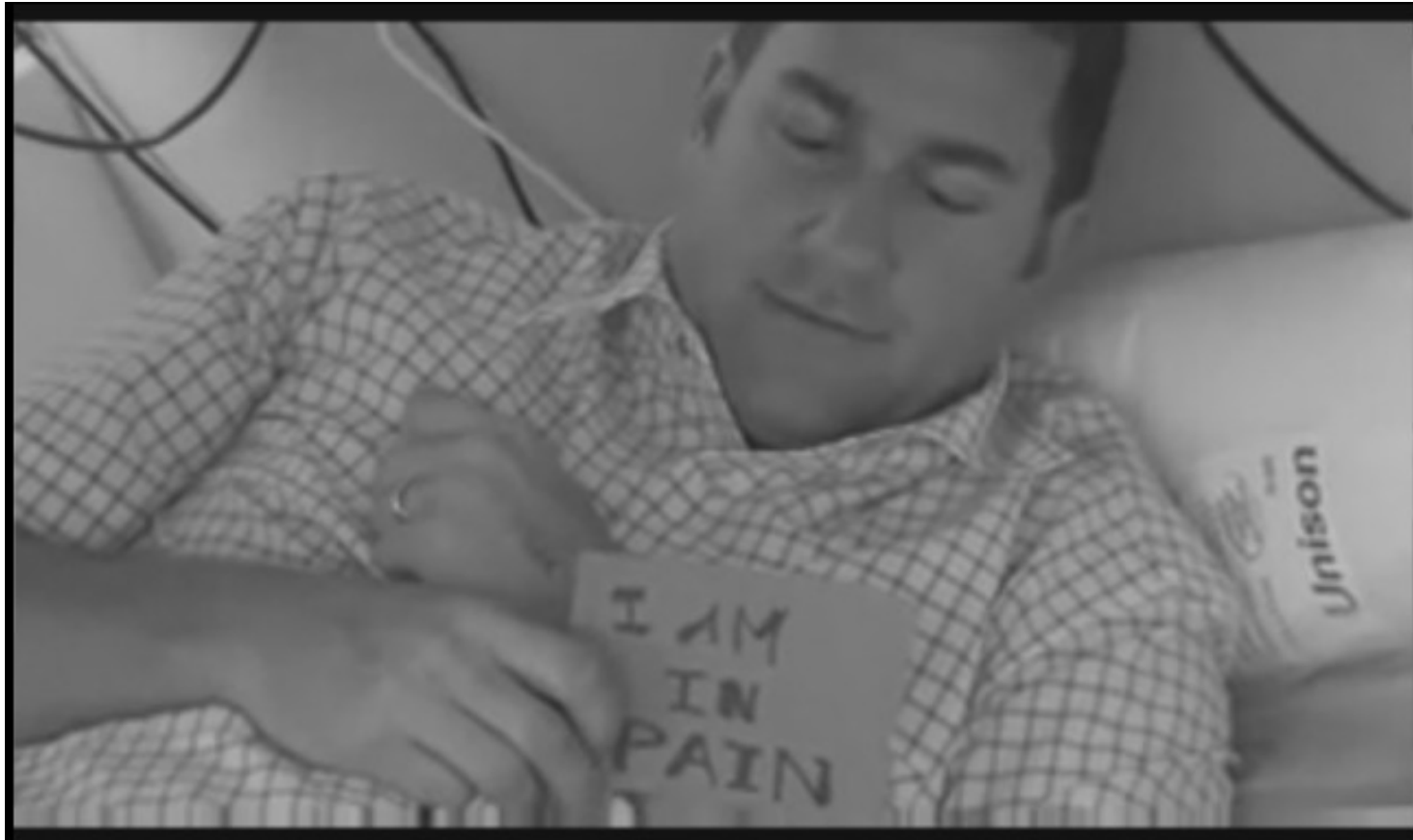


Desenvolvimento ágil de Software

- Motivação
 - Manifesto Ágil e Conceitos
 - SCRUM
 - *Extreme Programming (XP)*
-
- Slides serão disponibilizados. Anote apenas suas perguntas.



Motivação Interna – “Dor”



<http://www.youtube.com/watch?v=1lqxORnQARw>



1º Encontro PHPMG



Motivação Interna - *Chaos Report*

• Projetos de Pontes

- Prazo – OK (menos no Brasil)
- Orçamento – OK
- Quase nenhuma cai
- Ciência Antiga – 4 a 6 mil anos

• Projetos de *Software*

- Prazo – estoura
- Orçamento – estoura
- Têm problemas com frequência
- Ciência nova – 50 anos

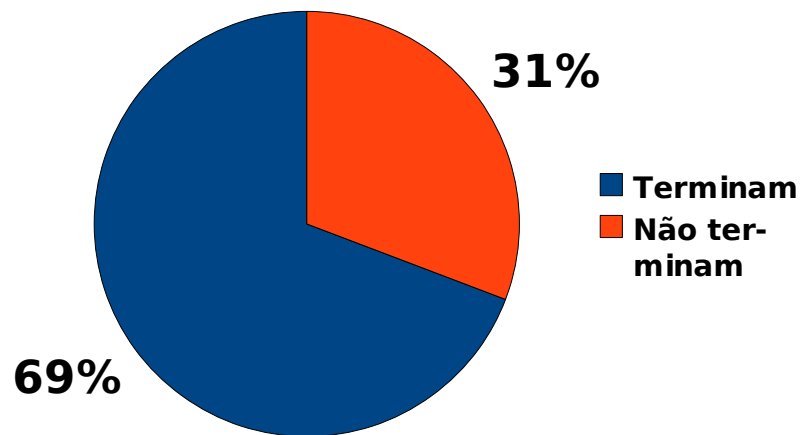
Aspectos críticos

- Projetos de ponte são engessados e ninguém dá “pitaco”
- Projetos de *software* normalmente precisam suportar mudanças nas regras de negócio
- Pontes que caem têm relatórios de erros. *Softwares* são mascarados e ignorados. Não há aprendizado



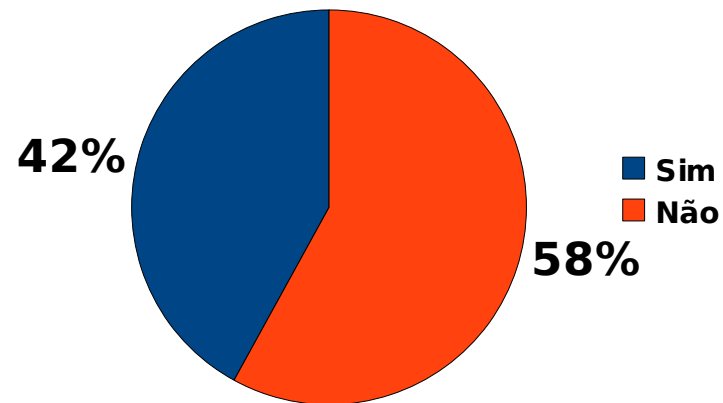
Motivação Interna - *Chaos Report*

Projetos que terminam



Motivação Interna - *Chaos Report*

Requisitos presentes ao final



Motivação do Mercado

- RAD – *Rapid Application Development* – anos 90.
 - Métodos iterativos (ciclos) e evolucionários (*bottom-up*)
- Empresas buscam produtos de TI como forma de diferenciação
- Frustração com planejamentos
- Necessidade de atendimento a modelos de maturidade
 - CMMI, MPS.BR
- Alternativas à época com baixa tolerância a mudanças de requisitos



E aí!?



1º Encontro PHPMG



E aí!?

- Desenvolvimento ágil não garante necessariamente que o projeto terá mais ou menos sucesso :-)



E aí!?

- Desenvolvimento ágil não garante necessariamente que o projeto terá mais ou menos sucesso :-(
• Mas ajuda!!! **Como?**



E aí!?

- Desenvolvimento ágil não garante necessariamente que o projeto terá mais ou menos sucesso :-(
 - Mas ajuda!!! Como?
 - Ajuda a descobrir antes que algo não está indo bem –
ITERAÇÕES CURTAS E ENVOLVIMENTO DO CLIENTE PARA VALIDAÇÃO
- :-))



Manifesto Ágil

- Encontro de 17 agilistas – Utah – Fevereiro – 2001
- Kent Beck – XP (*Extreme Programming*)
- Ken Schwaber – SCRUM
- Alistair Cockburn – Crystal – Metodologia sob demanda
- Chegaram a conclusões:
 - www.agilemanifesto.org



Manifesto Ágil

- **Individuals and interactions** over processes and tools
 - Uma descrição mínima de processo é necessária para se começar a trabalhar.
 - Cliente sempre presente



Manifesto Ágil

- **Working software** over comprehensive documentation
 - Software bem organizado e documentado
 - Alguma documentação existe. Apenas o suficiente e não conta como produto, resultado de trabalho.



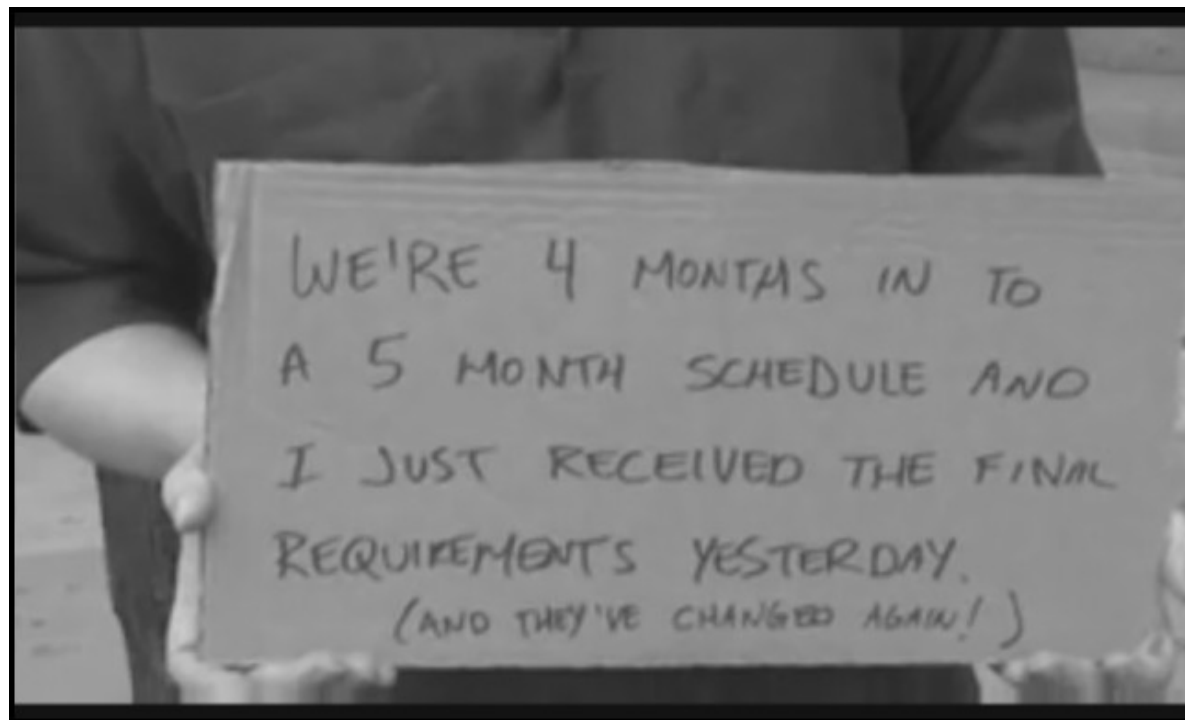
Manifesto Ágil

- **Customer collaboration** over contract negotiation
 - Cliente deve estar 'infiltrado' na equipe de desenvolvimento.



Manifesto Ágil

- Responding to change over following a plan



Método x Processos

- XP e SCRUM **não são processos**
 - Processos definem fluxo, entradas saídas, papéis.
- **São métodos (ou metodologias)**
- Esse entendimento facilita a adoção de práticas ágeis em processos tradicionais já definidos – **não precisa substituir o processo**
- Importante diferenciar também **processo de software** e **gestão de projeto de software**



SCRUM

- O nome vem do rugby. Reinício da partida.
- Baseado na teoria de controle de processo industrial
 - Auto-organização e emergência
- Utilizado há 15 anos com sucesso em milhares de projetos, centenas de organizações
- É gerencial. Não serve apenas para projetos de software

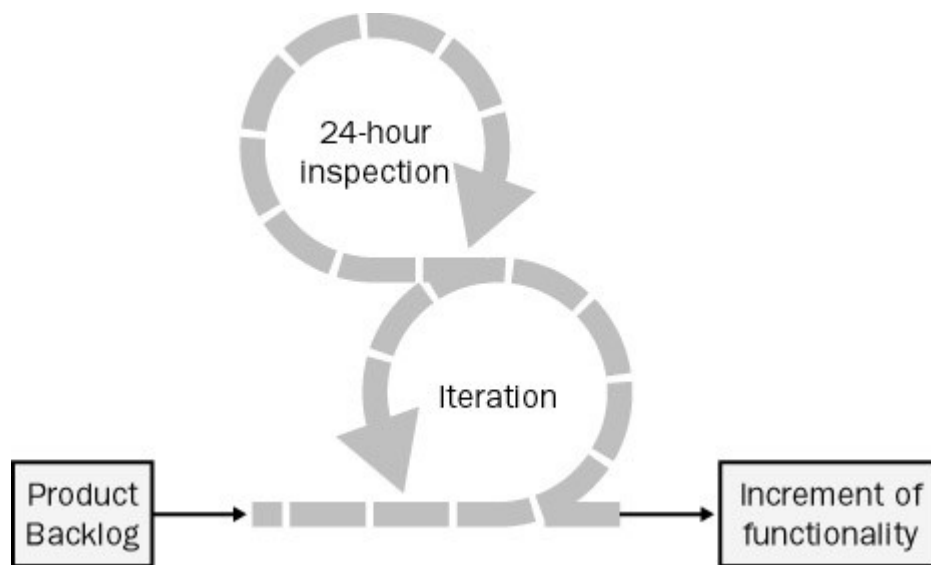


SCRUM

- Ajuda a controlar **projetos** de desenvolvimento de software
- Não garante **sucesso completo** do projeto
- Garante que o trabalho é dedicado aos resultados de **maior valor agregado**
 - Se o recurso for cortado, cliente sempre vai ter algo em mãos com alguma utilidade.
 - Requisitos importantes nunca ficam para o final



SCRUM



- Obtém-se do backlog o que é de mais valor
- Planeja-se a iteração
- Faz-se acompanhamento diário
- Entrega acréscimo de funcionalidades ao fim da iteração.

SCRUM - Papéis

- Product Owner (CLIENTE)
 - Lista de requisitos (*product backlog*)
 - Objetivos de ROI
 - Planejamento de Releases (priorizar)
- Team (EQUIPE)
 - Desenvolvimento de funcionalidades
 - Auto-gerido e auto-organizado (experiência)
 - Multi-funcional (programador, testador, arquiteto, etc)

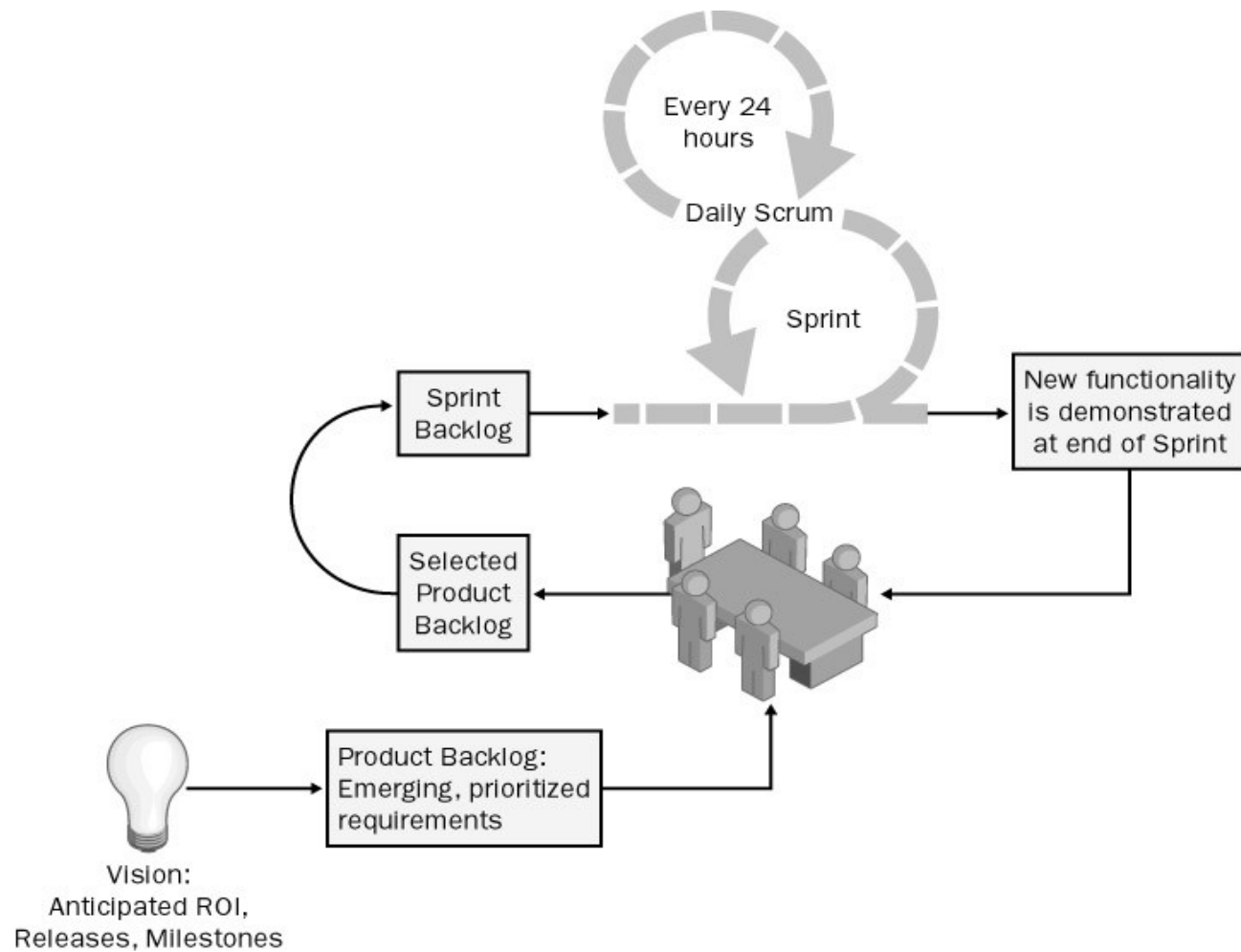


SCRUM - Papéis

- ScrumMaster
 - Ensinar Scrum aos outros envolvidos
 - Manter o método nos trilhos
 - **Respeitar cultura da organização x Entregar benefícios**
 - **CULTURA** é uma das principais barreiras a serem vencidas
 - Garantir que os outros envolvidos sigam as regras e práticas do SCRUM



SCRUM – Visão Geral



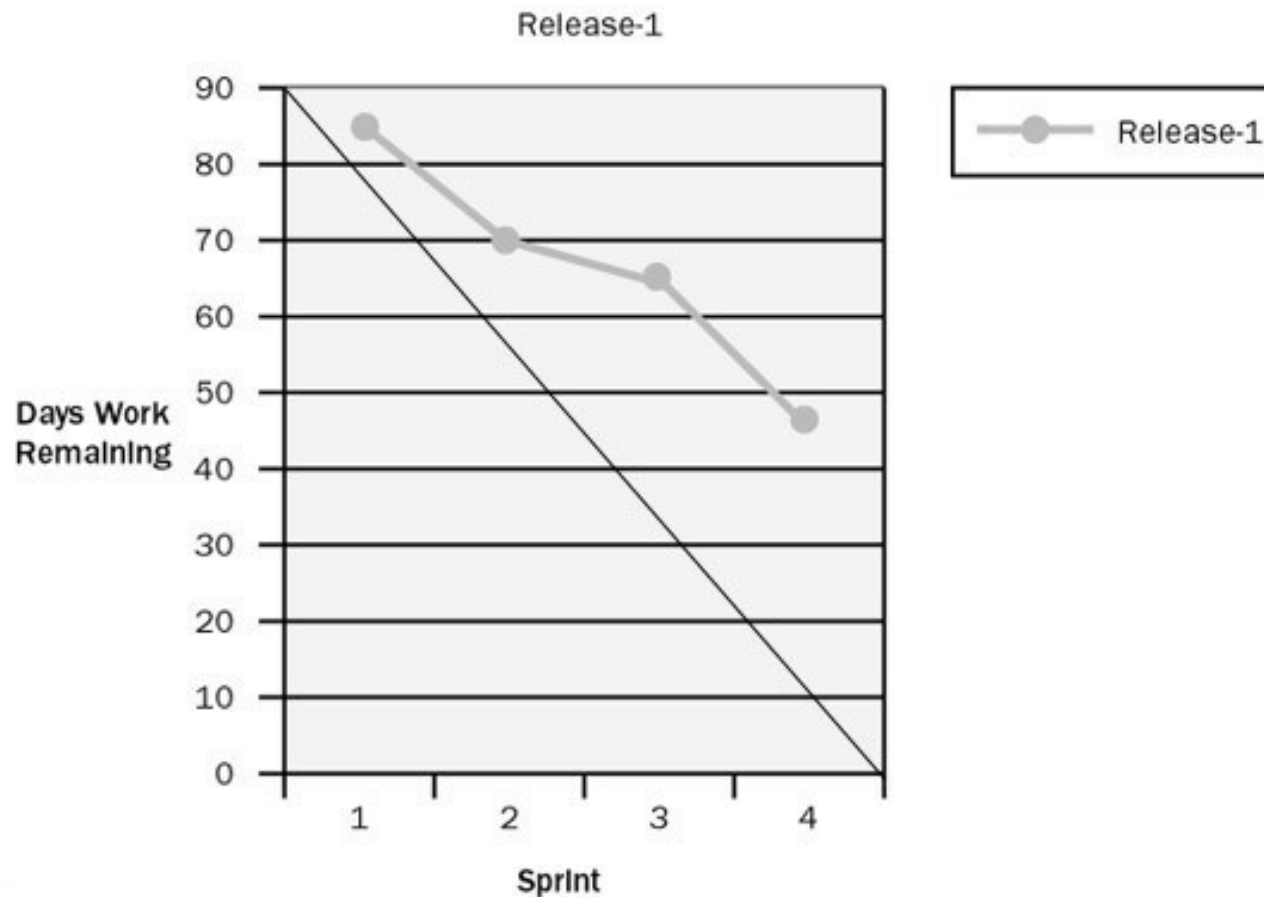
SCRUM - Artefatos

- Product backlog
 - Sempre evolui
- Sprint backlog
 - Derivado a partir do product backlog
 - Detalha os itens do product backlog em tarefas



SCRUM - Artefatos

- Burndown Chart – quanto mais horizontal, melhor



SCRUM - Artefatos

- Incremento de funcionalidade de produto potencialmente 'empacotável'
 - Esse incremento deve ser levantado em cada sprint
 - CLIENTE pode querer implantar (Antecipação ao release. Furo no SCRUM? Equipe estará apta?)
 - O que é um incremento **CONCLUÍDO?** (*done*)
 - Testado
 - Código bem escrito e bem estruturado
 - Disponível em um executável
 - Com documentação de usuário



SCRUM - Regras

- Sprint Planning Meeting – parte inicial – 4 horas
 - 4 horas definindo itens mais importantes e empacotáveis do backlog de produto
 - Todos os papéis participam
 - Backlog deve ser preparado antes pelo Product Owner(de preferência) ou ScrumMaster(pior)
- Sprint Planning Meeting – parte final – 4 horas
 - SCRUMMASTER responde perguntas da EQUIPE nas 4 horas finais para detalhamento de tarefas
 - Ao final, tem-se o Sprint Backlog



SCRUM - Regras

- Daily Scrum Meeting
 - 15 minutos independente do número de membros
 - Muita rigidez com presença e pontualidade
 - Três questões
 - O que você fez desde a última conversa?
 - O que você vai fazer de agora até a próxima?
 - O que lhe impede de fazer o seu trabalho o mais eficientemente possível?
 - Exigem respostas rápidas



SCRUM - Regras

- Sprint – duração sugerida: 30 dias
 - Itens de Backlog de sprint **CONGELADOS** durante a execução do sprint
 - Atendimento a mudanças de requisitos garantida pela continuidade de alterações no backlog de produtos
 - ScrumMaster pode abortar o sprint (casos extremos)
 - Team pode consultar ao P. Owner o que retirar do backlog quando for constatada impossibilidade de finalizá-lo por completo. O contrário (acrescentar funcionalidades) também é aceito.



SCRUM - Regras

- Sprint Review Meeting
 - 4 horas
 - Apresentar funcionalidades ao Cliente e *stakeholders*
 - Artefatos não podem ser apresentados como produtos de trabalho (Forma de policiar o contrato? Afinal o que tem valor é software funcional – valor ágil)
 - Stakeholders são ouvidos
 - Ao final, o próximo Sprint Review Meeting é agendado



SCRUM - Regras

- Sprint Retrospective Meeting
 - 3 horas
 - SM, TM e PO (opcional)
 - Perguntas ao TM
 - O que foi bom no último sprint?
 - O que não foi bom?
 - Melhorar práticas
 - SM cataloga as respostas
 - TM prioriza a ordem de melhoras em potencial para discutir



Gostaram do SCRUM?

- Falamos de **Gestão de Projeto**
- Agora 'tá' na hora de práticas de desenvolvimento
- Vamos falar de XP



XP – Extreme Programming

- “Jeito leve, eficiente, de baixo risco, flexível, previsível, científico e divertido de desenvolver software” - Kent Beck
- Você falaria de outro jeito do seu método???



XP – Extremo

- Extremo porque leva os princípios e práticas de senso comum ao extremo
 - “Se revisar código é bom, **nós revisaremos código o tempo todo** (programação em pares)”
 - “Se testar é bom, **vamos testar o tempo todo** (testes de unidade), inclusive os clientes (testes funcionais e de aceitação)”
 - São 12 práticas na primeira versão. Já foi estendido.
- Conservador: a maioria das práticas existem há décadas



XP – Promete

- Aos programadores que eles **trabalharão todos os dias com coisas que realmente importam** (entre outras coisas)
- Aos clientes e gerentes que eles **terão o maior valor agregado possível** a cada semana de desenvolvimento (entre outras coisas)



XP – Valores

- Comunicação
 - Interna e com o cliente
- Simplicidade
 - Não olhar para frente. Para que pensar no trabalho de amanhã, se amanhã talvez ele não faça mais sentido?
 - “XP is making a bet. It is betting that it is better to do a simple thing today and pay a little more tomorrow to change it if it needs it, than to do a more complicated thing today that may never be used anyway.” - **Kent Beck**



XP – Valores

- Feedback
 - Do estado do sistema. Testes.
 - Do cliente. Testes funcionais.
 - Da equipe para o cliente quando se planeja as liberações
- Coragem
 - Desenvolver sem pensar no que pode acontecer amanhã não é um ato de coragem?
 - Jogar código fora também é, certo?



XP – Algumas Práticas

- Programação em pares
- Jogo do Planejamento e *planning poker*
- TDD
- Integração contínua
- Pequenas liberações
- *Refactoring*
- Padrões de Código e código de vários donos



XP – Algumas Práticas

- Programação em pares
- Jogo do Planejamento e *planning poker*
- TDD
- Integração contínua
- Pequenas liberações
- *Refactoring*
- Padrões de Código e código de vários donos
 - **XP vai além da gerência!!!**



XP – Algumas Práticas

- Jogo do Planejamento e Planning poker



XP – Algumas Práticas

- 40 horas por semana e pessoal motivado
 - “Contrate bons profissionais, dê a eles ferramentas e treinamentos para que façam o trabalho, depois saia da frente deles” Dave Thomas – OBT
 - “Dê café também” - Sabedoria popular



Pontos críticos para implantação de métodos ágeis

- Escopo variável
- Cultura da organização
- Presença do cliente
- Membros devem ser experientes e envolvidos
- Melhoramento de processo – repetir o que é bom e descartar o que não é
- Eleger um bom ScrumMaster **que não programe**
- Obter credibilidade diante das organizações



Situações favoráveis à adoção de métodos ágeis

- Equipes pequenas e centralizadas
- Programadores experientes
- Cliente sem requisitos formalizados
- Requisitos podem mudar mesmo quando o cliente sabe o que quer
- Orçamentos e prazos limitados
- Mais alguma situação?



Ferramentas de apoio à gestão ágil de PROJETOS

- Os "agilistas" não gostam muito de utilizar ferramentas, pois elas tiram agilidade.
- Exemplos Free
 - Post-it, IceScrum2, Agilefant
 - Xpweb (única em PHP), Xplanner
- Exemplos Comerciais
 - Mingle, Target Process (gratuita para até 5 usuários)



Agilidade x CMMI

- O próprio SEI – *Software Engineering Institute* - incentiva em um relatório técnico
- É onde estou tentando contribuir
 - Adição de agilidade em áreas de processos do CMMI relacionadas ao grupo *engineering*.
 - Atender às recomendações do CMMI com práticas ágeis
- Há esforços (brasileiros) de extensão do SCRUM para atendimento ao CMMI – áreas de processo relacionadas a gestão de projeto



Há treinamentos e certificações

- Certificação ScrumMaster
 - Muito contestada
 - Não há empresas em BH fazendo
 - TEAMWARE e CAELUM em SP
- Treinamentos
 - Há empresas em BH oferecendo para SCRUM especificamente.
 - **Ver apenas SCRUM pode ser incompleto**
 - Melhor aproveitado após esforço interno.



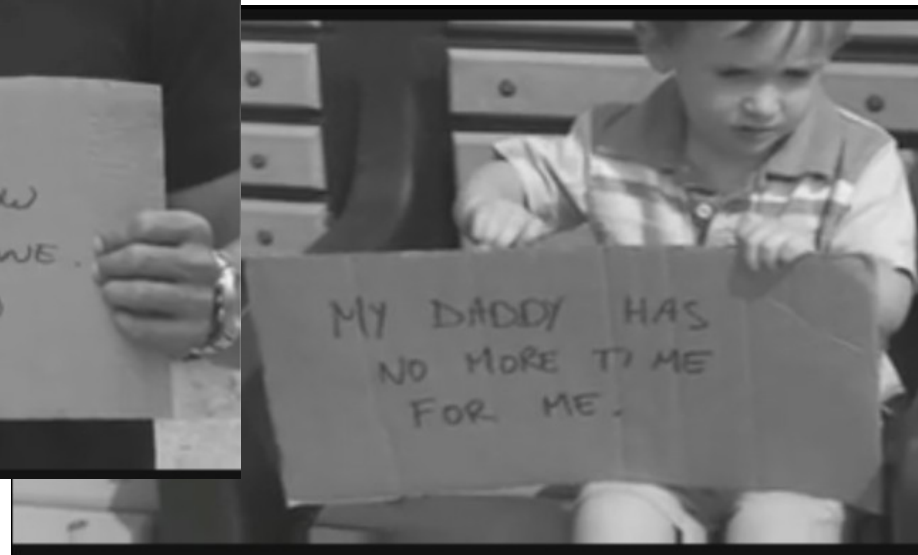
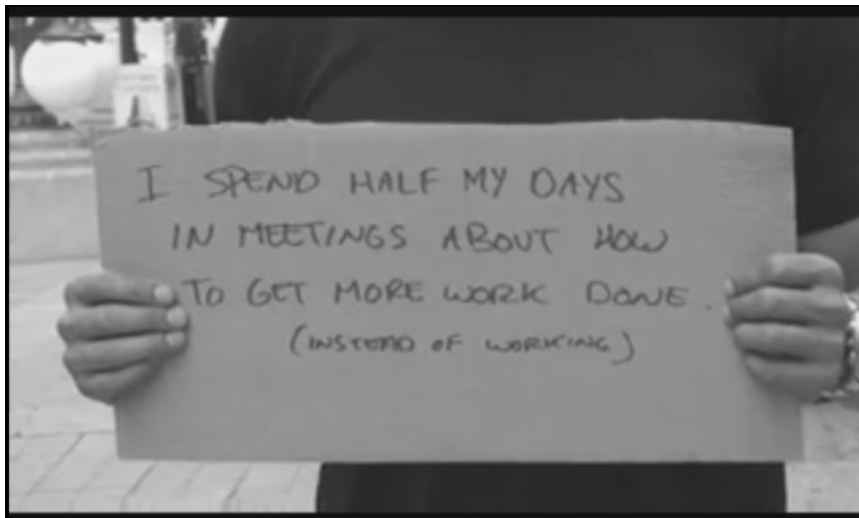
Há treinamentos e certificações

- Cursos relacionados da PHPPrime
 - Automação de testes para WEB (parceria Base2)
 - Desenvolvimento Ágil de Software



Obs.:

- Idéia(?) original de usar o vídeo Hug a Developer Today:
 - Dairton Bassi – IME/USP



Acabando...

- Dúvidas, críticas, sugestões?
 - F6 Sistemas | PHPPrime Training
 - joaquim@f6sistemas.com.br
 - (31) 3785- 1130 | (31) 9130-1042
- Ajudem a divulgar:
 - agilminas@googlegroups.com
 - agilbr@googlegroups.com

