



## Acompanhamento ágil

Baseada na apresentação do LABXP e na dissertação de Renan Oliveira

Adaptação nos slides de Viviane Santos  
Instituto de Matemática e Estatística - USP

# Agenda

---

- IME/USP: Laboratório XP e Grupo Soma
- Acompanhamento ágil – Conceitos
- Exemplos do LabXP 2011 (IME/USP)
- Guidelines

- ❑ Métodos ágeis e métricas de software
- ❑ Desde 2011
- ❑ Integrantes: 3 professores, 4 alunos de doutorado e 6 alunos de mestrado
- ❑ Reuniões todas as sextas-feiras
- ❑ Objetivo: Discussão de temas relacionados e colaboração em pesquisas conjuntas

# Acompanhamento ágil - Conceitos

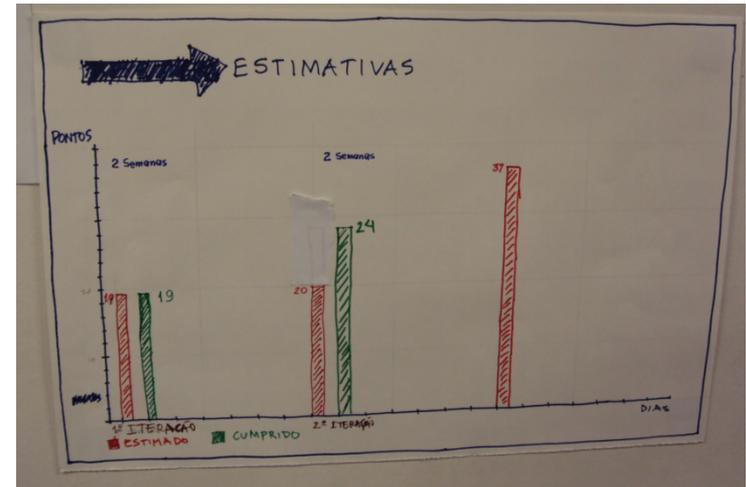
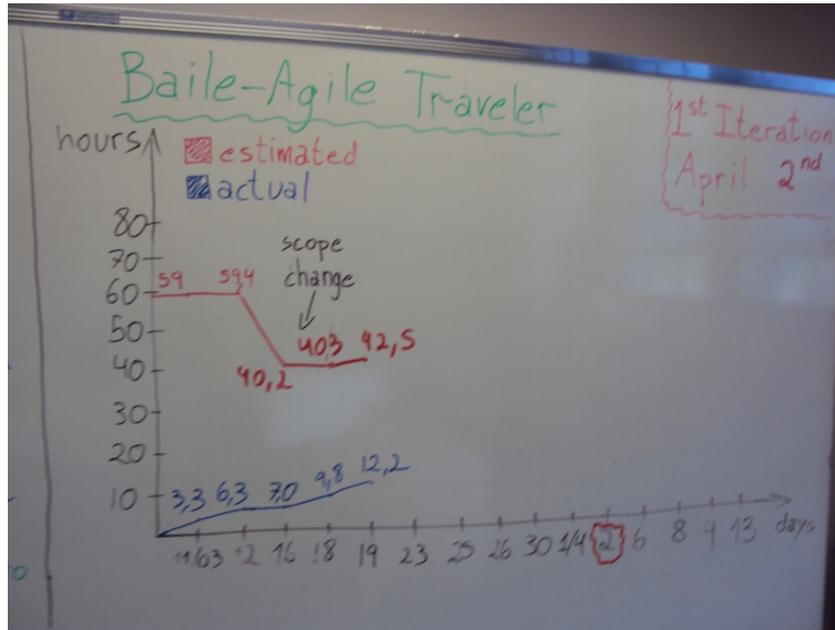
# O que é?

- ❑ Manipulação e disponibilização informal de métricas e informações úteis no contexto do projeto
- ❑ Utiliza-se recursos visuais no próprio ambiente de trabalho
- ❑ Tarefas: análise, escolha, manipulação e disponibilização de métricas/informações úteis



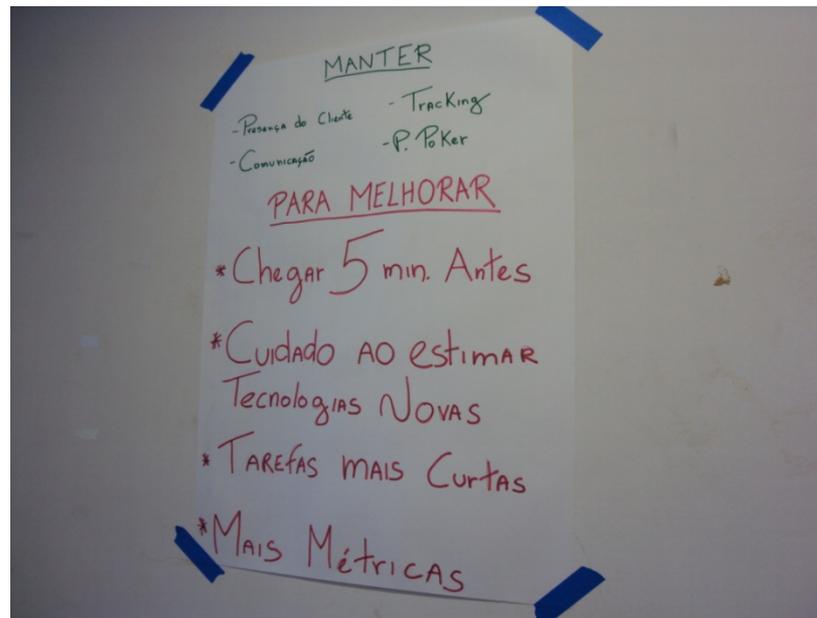
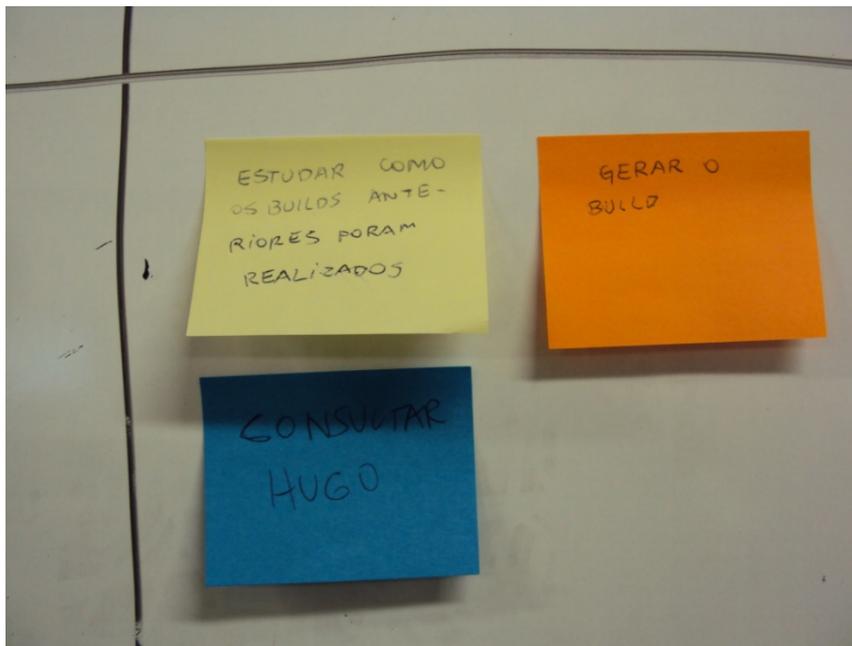
# Exemplos

## □ Gráficos



# Exemplos

## □ Lembretes



# Exemplos

## □ Ferramental



# Exemplos

## □ Outras informações

CoGroo  
Comunidade

EMANUEL KON-ECL

	ROTOR	HUBER	INFORMAS	BOGROD	XM/XSM	UNIT
Eduardo	☹	☹	☹	☹	☹	☺
Fillipe	☹☹	☹☹	☹☹	☹☹	☹☹	☹☹
Robson	☹☹	☹☹	☹☹	☹☹	☹☹	☹☹
Thiago	☹☹	☹☹	☹☹	☹☹	☹☹	☹☹
Wesley	☹☹	☺☺	☺☺	☹☹	☹☹	☺☺
William	☹☹	☹☹	☹☹	☹☹	☹☹	☹☹

Matriz de competência

☹☹ o que é?  
☹☹ Acha que sei!  
☺☺ Guru

Dificuldade

programação extrema

CONCEITO DE PRONTO DE UMA HISTÓRIA

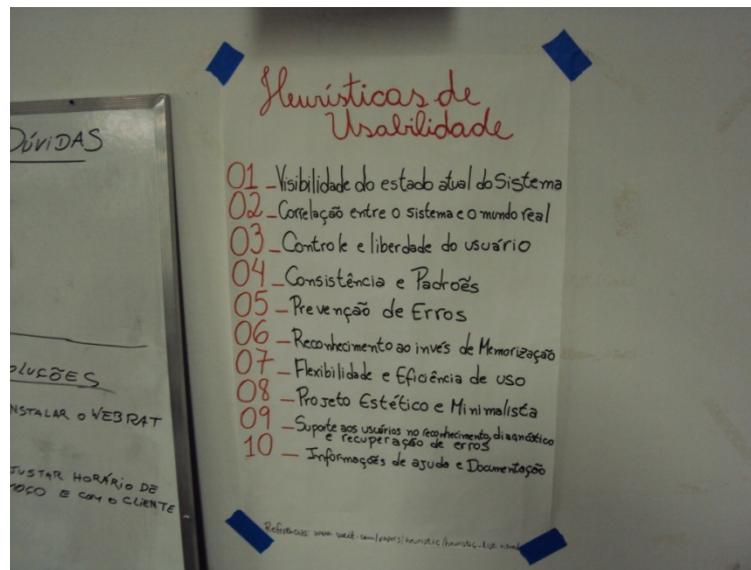
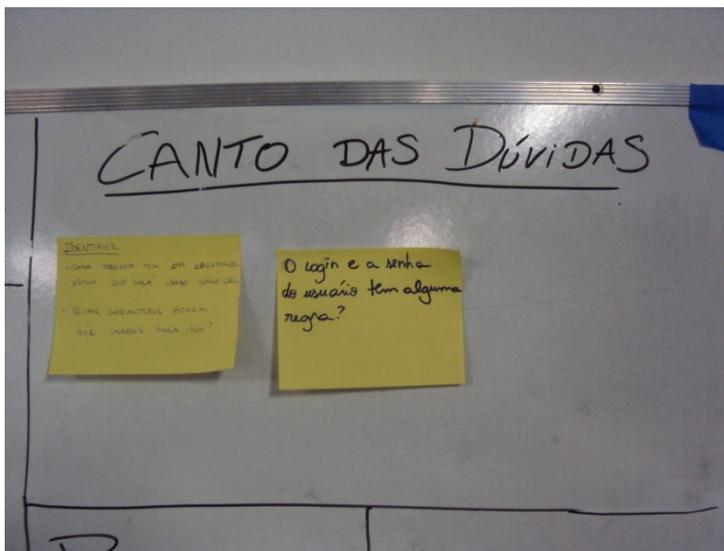
- TODAS AS TAREFAS DA HISTÓRIA EXECUTADAS
- CÓDIGO SEM ERROS DE COMPILAÇÃO E SOB CONTROLE DE VERSÃO
- COM 100% DOS TESTES DE UNIDADE PASSANDO
- COM COBERTURA DE TESTES  $\geq$  \_\_\_%

Planning Feedback Loops

IMPED

# Exemplos

## □ Outras informações

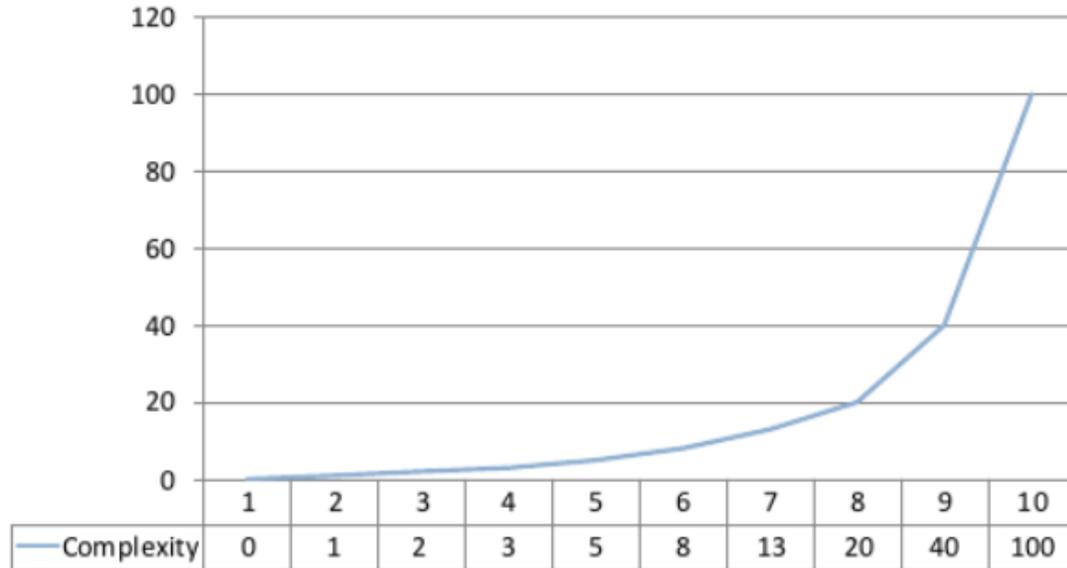


# Princípios ágeis no acompanhamento

- Feedback
- Adaptabilidade
- Humanidade
- Foco e fluxo de trabalho
  - ▣ Quais itens foram completados desde a ultima reunião
  - ▣ Quais impedimentos foram encontrados que precisam ser resolvidos
  - ▣ Quais novas tarefas a equipe deve realizar até a próxima iteração
- Comunicação, reflexão e aprendizado
- Interação com o cliente

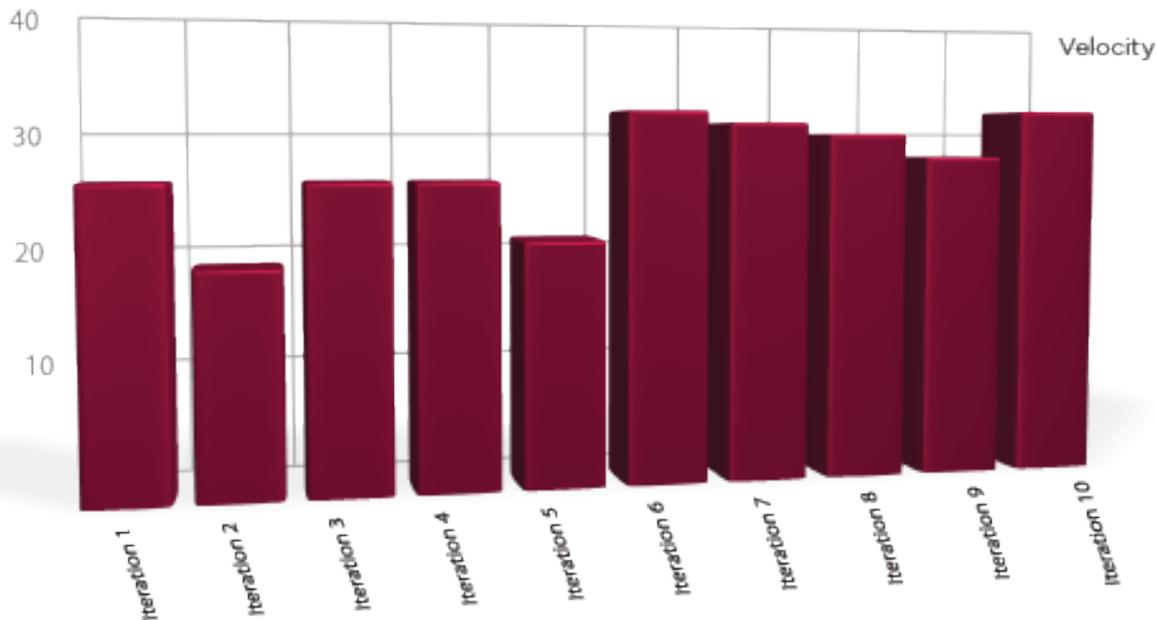
# Estimativa

- ❑ Previsão + incerteza
- ❑ Refinamento
- ❑ Pontos
  - ▣ Medida pura de tamanho
  - ▣ Estimativa relativa
  - ▣ Mais abstrato
  - ▣ Medida por uma escala (Ex: sequência de Fibonacci)
- ❑ Dias, dias ideais, horas



# Velocidade da equipe

- Soma das estimativas das funcionalidades entregues por iteração
- Tende a estabilizar com o tempo
- Base para planejamentos
- Curiosidade: equipes mais maduras geralmente deixam de coletar



Exemplo de gráfico de velocidade (Fonte: VersionOne)

# Outros

- ❑ Cobertura de testes
- ❑ Defeitos (incidências ou erros)
- ❑ Tempo das reuniões diárias
- ❑ Funcionalidades entregues por iteração
- ❑ Tempo de entrega de uma funcionalidade
- ❑ Estimado versus Realizado
- ❑ Revezamento de pares
- ❑ Humor da equipe
- ❑ Satisfação do cliente

# eXtreme Programming (XP)

- Espaço de trabalho informativo
  - ▣ Capturar o andamento geral do projeto em 15 segundos
  - ▣ Suprir algumas necessidades humanas
  - ▣ Identificação de problemas reais ou potenciais algum tempo
- Papel do *tracker*
  - ▣ Coleta de dados referentes às métricas
  - ▣ Consciência da equipe
- Acompanhamento reflete a capacidade de aprender
- “Big visible Charts” de Ron Jeffries (<http://xprogramming.com/articles/bigvisiblecharts/>)
- Deve ter um equilíbrio: pode consumir tempo e desestimular a equipe

# Crystal

- ❑ Família de metodologias
- ❑ Crystal Clear para equipes de até 8 integrantes
- ❑ Entrega frequente, comunicação próxima e otimização reflexiva
- ❑ Radiadores de informações
  - ▣ Devem ser grandes, visíveis, compreensíveis, atualizáveis e ser de valor constante para a equipe
- ❑ Comunicação osmótica

# Lean Software Development

- Adaptação da abordagem de desenvolvimento de produtos da manufatura
- 7 princípios: “Eliminar desperdício”, “Amplificar o aprendizado”, “Decidir o mais tarde possível”, “Entregar o mais rápido possível”, “Fortalecer a equipe”, “Construir com integridade”, “Ver o todo”
- Acompanhamento: aprendizado contínuo
- Trabalho auto-direcionado necessita de controles visuais
- Listas de problemas encontrados, idéias para melhoria, candidatos para refatoração, etc.

# Kanban

- Sistema de “puxar”
- Compartilha informações:
  - ▣ Tarefas
  - ▣ Tipos de tarefas
  - ▣ Estado das tarefas
  - ▣ Estimativas
  - ▣ Atribuições

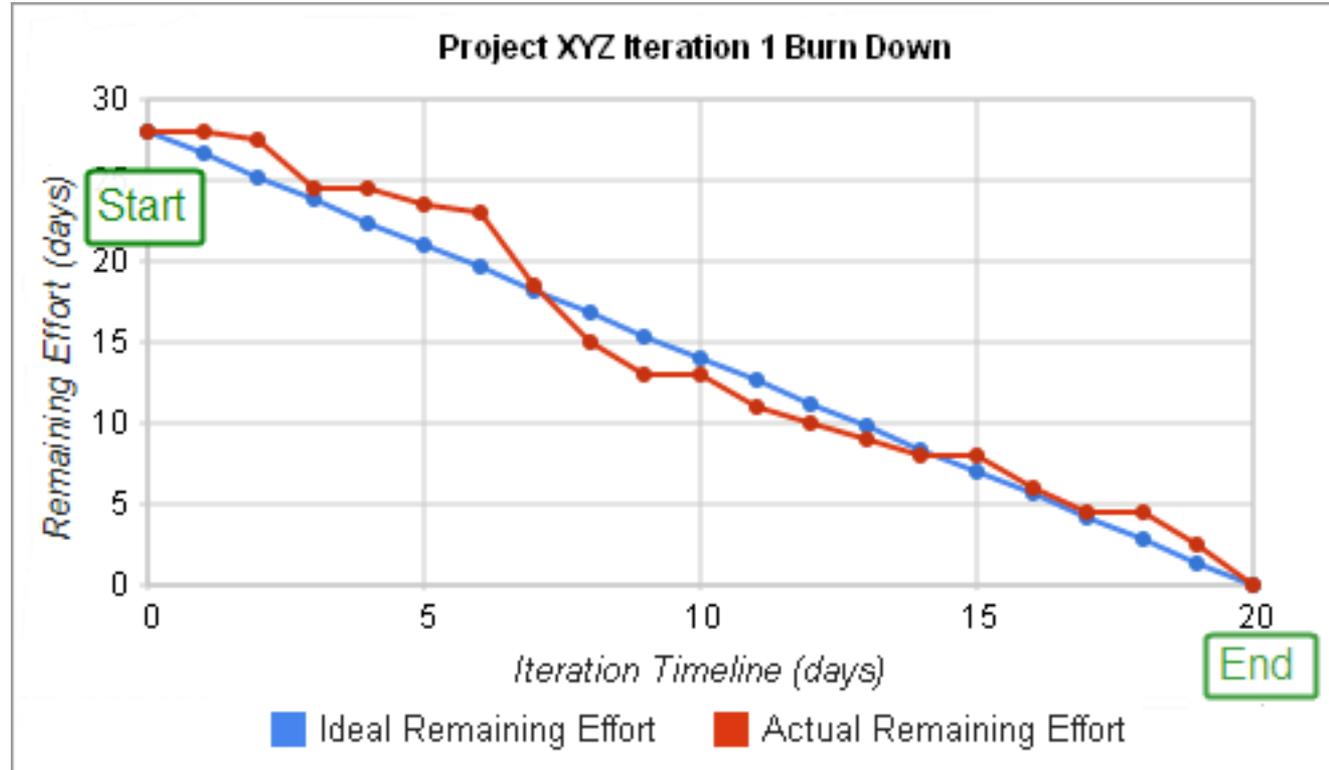


Fonte: Espaço informativo de um projeto do LabXP (2011)

# Scrum

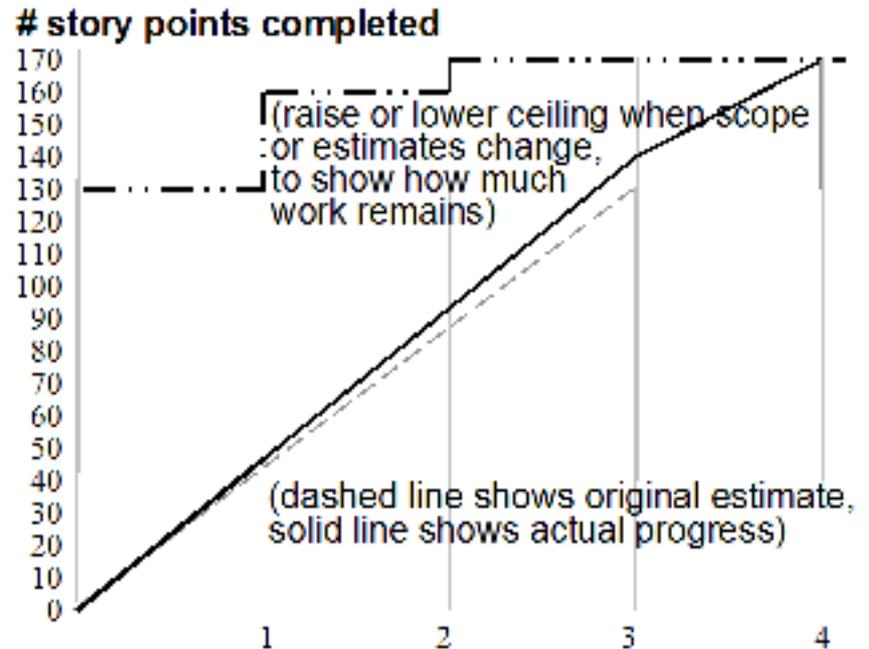
## □ Burndown

- ▣ Progresso da equipe em alcançar a meta da iteração
- ▣ Qtde trabalho restante em cada ponto do tempo
- ▣ Poder emocional
- ▣ Não é bom para lidar com mudanças de escopo



# Scrum

- Burn-up
  - ▣ Planejado x Realizado
  - ▣ Mostra também o impacto das mudanças em relação a data da entrega



# Outros indicadores

- Calendário Niko-Niko
  - Humorômetro
  - Ajuda a sinalizar problemas na equipe/ projeto

Anderson - 1

Motivação / Moral (por dia)

	19/06	22/06	26/06	29/06
ANDERSON	☹️ / -	☹️ / ☹️	😊 / ☹️	😊 / 😊
CRIS	castigo ☹️	😊 / 😊	😊 / ☹️	☹️ / ☹️
Filipe	😊 / 😊	😊 / 😊	castigo ☹️	falta
Gigante	😊 / 😊	😊 / 😊	☹️ / ☹️	castigo ☹️
Hugo	☹️ / ☹️	😊 / 😊	castigo	😊 / 😊
MARCIÔ	☹️ / 😊	😊 / 😊	😊 / 😊	😊 / ☹️
MARI	☹️ / ☹️	😊 / -	☹️ / ☹️	☹️ / ☹️
Sensei	😊 / ☹️	castigo	😊 / ☹️	😊 / 😊

← HUMORÔMETRO →

# Outros indicadores (cont.)

## □ Revezamento de pares

	João	Ana	Carol
Marcos	1	6	2
Carol	7	1	
Ana	3		

# Sinalizadores

## □ Lava lamp



## □ Até música!

## □ Faróis



# Exemplos do LabXP 2011

# Exemplos de quadros

Semnt 2  
03/06 → 22/06

**TRACKER: PEDRO HUBO**

**Parcamento**

	H	P	M
J	□	□	□
M	□	□	
P	□		

**Todo** | **Dev** | **Pronto**

**COMMENTS**

criar campo revisado

integrar denúncia dentro do comentário

criar comentário na denúncia

criação de lista de denúncias

criar comentário na denúncia

**Lembra**  
flore - slides CSS  
↳ .di - sl: de L ... slides.html

2 semanas

**NEWS!**

**Historia de Menções**

1 - CONFIG. SISTEMA  
- SEQUEL CRIODRAGAM  
- ESTUDO DA PLATAFORMA

2 - DEJUVIAS (TARA MOCERAVO)  
ROBERTO CONFIRMAR DEJUVIA DEJUVIA 17

3 - REPORTAR ABUSO 17

4 - ESTUDAR RATING (COMPOENTE) 12

**TILES**

OCULTAR CONTEUDO BASEADO EM REGRES

AR 17

ATUALIZAÇÃO DE SENHA

104

103/06

08/06

10/06

15/06

19/06

• ESPERADO X EFETIVO

GANHO

BLUE SCREEN

22/06: Relo, Dom castigado  
21/06: Merge da front

CODE FREEZE

# ARCHIMEDES - TRACKING BOARD

PERÍODO: 5ª iteração  
 DATA: 22/06 ~ 29/06

EMAIL DO ADMIN DA REDE:  
 KON-ECLIPSE-ADMIN@IME.USP.BR  
 Telefones de EMERGÊNCIA  
 Nelson: (11) 9459-5449  
 Geraldo: (11) 8452-2372

- Fizemos:
- Correção do build
  - Testes de interface
  - Testes unitários *cl* um click (RunAllTests)
  - Polígono regular
  - Elipse
  - Trim & Offset
  - Edit text
  - Importar DXF

- Legado:
- Intersectors da Elipse
  - Salvar em DXF
  - Importar MTEXT do DXF
  - Projection da Elipse

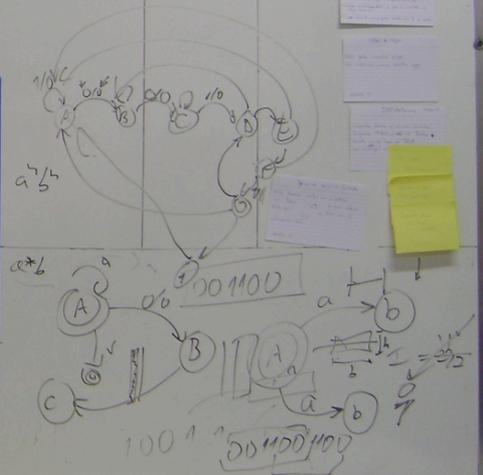
Parceramentos  
 Felipe Tagores TB Gustavo Chico

Polí	7	3	4	5	8
Chico	5	7	3	3	
Gustavo	3	3	8		
TB	4	7			
Tagores	8				

HUMORÔMETRO  
 Motivação: 😊😊😊  
 Produtividade: 😊😊😊

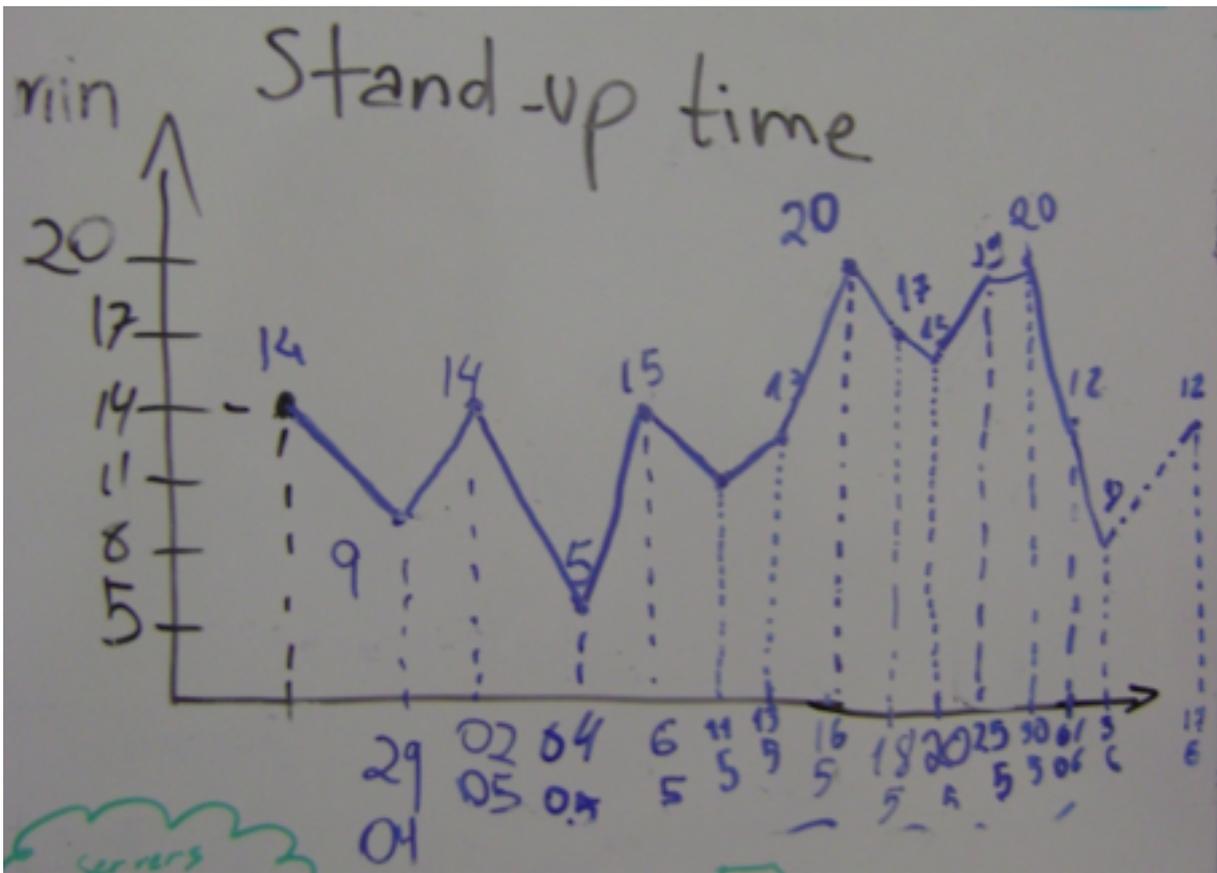
Nome	1	2	FINAL
Polí	RIP	☐/☐/☐	
Chico	☐/☐/☐/☐		
Gustavo	☐/☐/☐	RIP	
TB	☐/☐/☐/☐		
Tagores	RIP	☐/☐/☐	
Felipe	☐/☐/☐	☐/☐	

Todo Process Done (Dev+Tests)



Handwritten notes and sticky notes on the right side of the whiteboard, including a yellow sticky note.

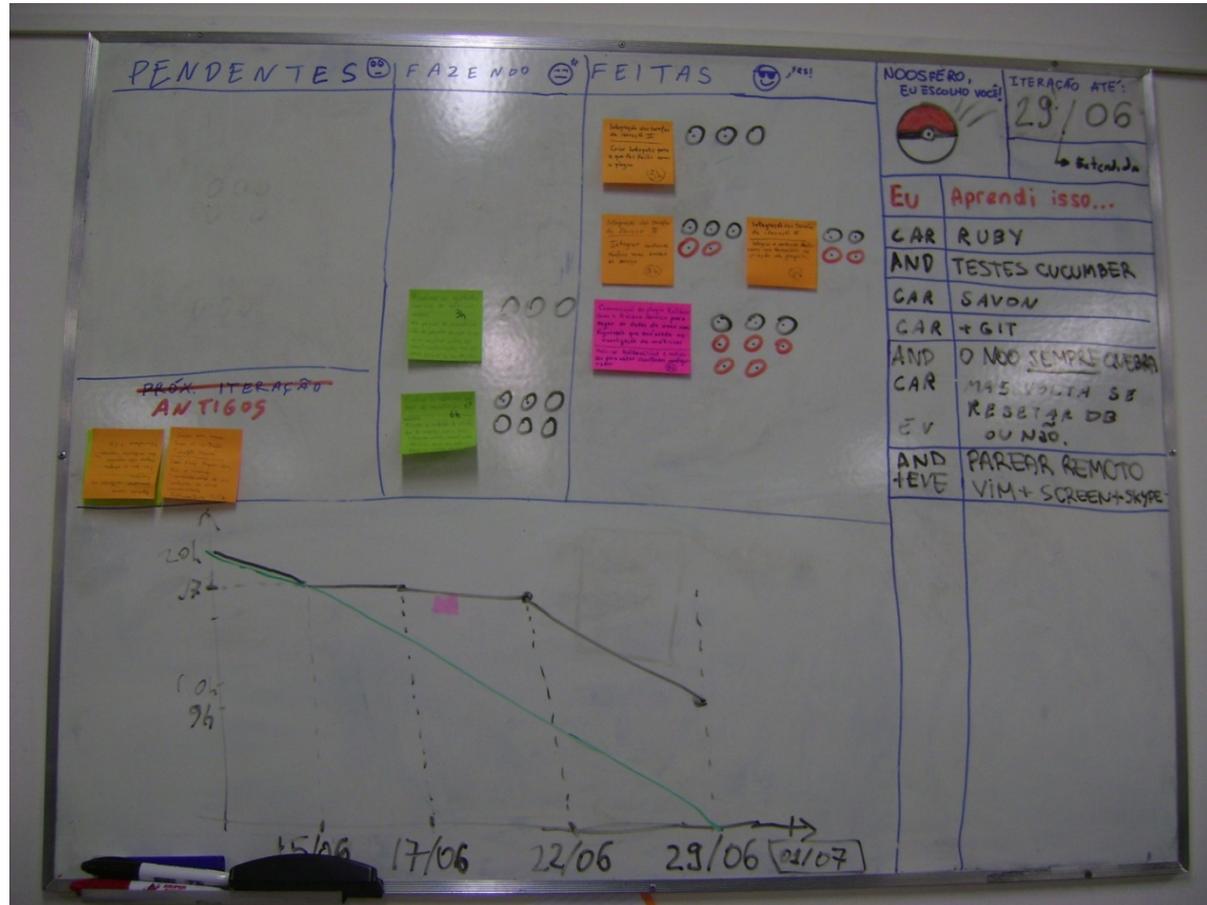
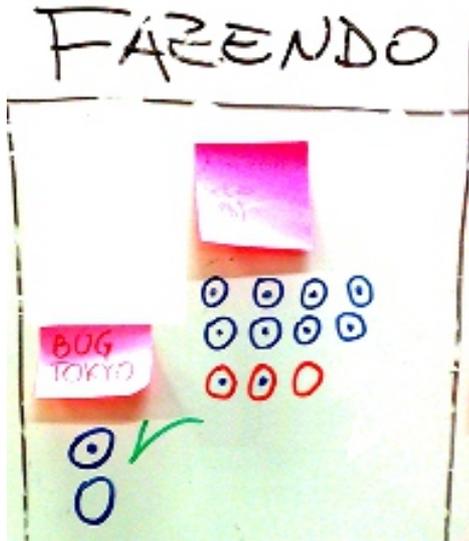




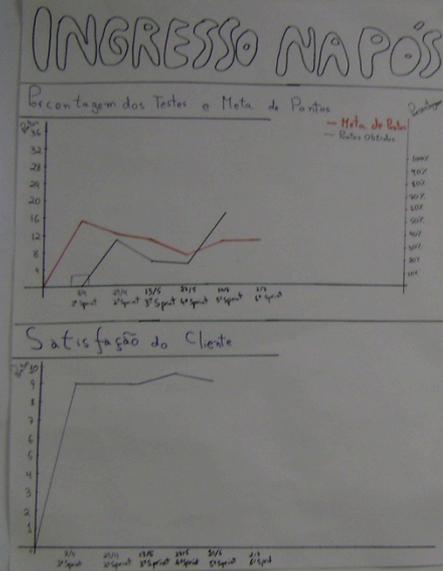
# Uso de Pokebolas

Fonte: <http://cemshost.com.br/bcc/w/LabXP>

- Estimativas inicial e final; horas trabalhadas; status das tarefas
- Ajuda a identificar erros de estimativa



# Exemplos de cartazes



## Planejamento

	Ana Luiza	Ashida	Diana	Gabriel	Renata	Thiago	Wilson	William
Wilson								
Wilson								
Thiago								
Renata								
Gabriel								
Diana								
Ashida								
Ana Luiza								

## Boas práticas de Git

commit e merge

```

git pull atualizar atualizações
git checkout -b <nome> cria uma branch local

git status cheia a branch e arquivos alterados
git add <arquivo> adiciona
git rm <arquivos> deleca
git commit

git checkout master volta de branch
git pull
git rebase master <nome branch> coloca os novos desenvolvimento na branch local

se tiver problemas de merge
git mergetool editor de merge
git rebase --continue continua o rebase

git checkout master
git merge <nome branch> junta na branch principal
git push upload
    
```

## Contato com o cliente

	Visitas	E-mail	Outros
06/05	0	1	0
13/05	0	2	0
20/05	1	0	0
27/05	0	0	0
03/06	1	0	0
10/06	1	0	0
17/06	0	0	0
24/06	0	0	0

## Mais comandos do Git

```

# voltar para um commit
git checkout commit_id

# deletar um branch local
git branch -d nome-da-branch

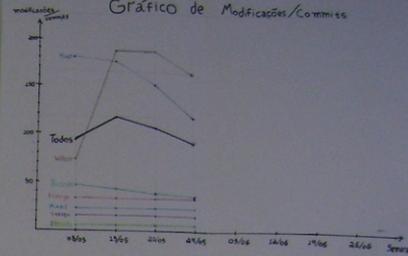
# volta para o último commit seu
git reset --hard

# pegar uma branch
git pull origin branch

# mandar uma branch local
git push origin minhaBranch

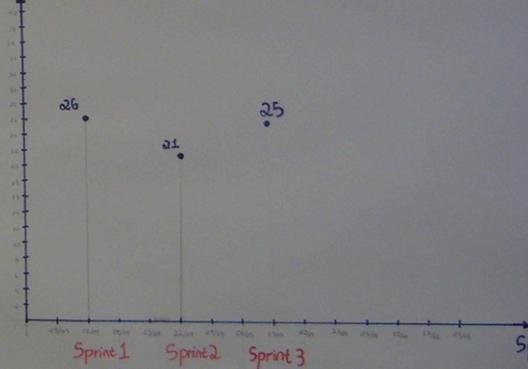
# para remover
git push origin :minhaBranch
    
```

## Gráfico de Modificações/Commits



## GRÁFICO DE PONTOS

Nº de Pontos



# Git Cheat Sheet

<http://git.or.cz/>

Remember: `git command --help`

Global Git configuration is stored in `$HOME/.gitconfig` (`git config --help`)

## Create

From existing data  
`cd ~/projects/myproject`  
`git init`  
`git add .`

From existing repo  
`git clone --existing/repo --new/repo`  
`git clone git://host.org/project.git`  
`git clone ssh://you@host.org/proj.git`

## Show

Files changed in working directory  
`git status`

Changes to tracked files  
`git diff`

What's changed between \$ID1 and \$ID2  
`git diff $ID1 $ID2`

History of changes  
`git log`

History of changes for file with diffs  
`git log -p $file $dir/ectory/`

Who changed what and when in a file  
`git blame $file`

A commit identified by \$ID  
`git show $id`

A specific file from a specific \$ID  
`git show $id:$file`

All local branches  
`git branch`

## Cheat Sheet Notation

\$id - notation used in this sheet to represent either a commit id, branch or a tag name  
 \$file - arbitrary file name  
 \$branch - arbitrary branch name

## Concepts

### Git Basics

master - default development branch  
 origin - default upstream repository  
 HEAD - current branch  
 HEAD~n - the great-great-grandparent of HEAD

### Revert

Return to the last committed state  
`git reset --hard`

Revert the last commit  
`git revert HEAD`

Revert specific commit  
`git revert $id`

Fix the last commit  
`git commit -a --amend`

Checkout the \$id version of a file  
`git checkout $id $file`

### Branch

Switch to the \$id branch  
`git checkout $id`

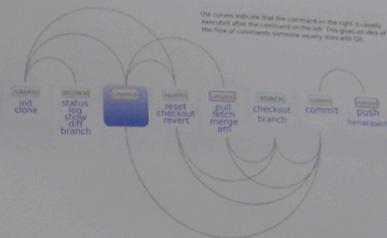
Merge branch1 into branch2  
`git checkout $branch2`  
`git merge branch1`

Create branch named \$branch based on the HEAD  
`git branch $branch`

Create branch \$new\_branch based on branch \$other and switch to it  
`git checkout -b $new_branch $other`

Delete branch \$branch  
`git branch -d $branch`

## Commands Sequence



## Update

Fetch latest changes from origin  
`git fetch`

Pull latest changes from origin  
`git pull`

Apply a patch that some sent you  
`git am -3 patch mbox`

Apply a patch that you made  
`git am --resolved`

## Publish

Commit all your local changes  
`git commit -a`

Prepare a patch for other developers  
`git format-patch origin`

Push changes to origin  
`git push`

Mark a version / milestone  
`git tag v1.0`

## Useful Commands

### Finding regressions

`git bisect start`  
`git bisect good $id`  
`git bisect bad $id`  
`git bisect bad/good`  
`git bisect visualize`  
`git bisect reset`

### Check for errors and cleanup repository

`git fsck`  
`git gc --prune`  
 Search working directory for foo!  
`git grep "foo!"`

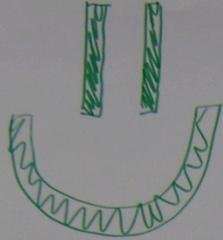
## Resolve Merge Conflicts

To view the merge conflicts  
`git diff`  
`git diff --base $file`  
`git diff --ours $file`  
`git diff --theirs $file`

To discard conflicting patch  
`git reset --hard`  
`git rebase --skip`

After resolving conflicts, merge with  
`git add $conflicting_file`  
`git rebase --continue`





Programamos  
mais do que  
na 1ª iteração

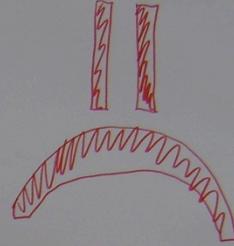
- ↳ Não pode Atalá
- ↳ Usar comunidade a nosso favor

~~Presidente~~  
Meta-Cliente  
ir na azul

- ↳ Mostrar ainda mais o que estamos fazendo
- ↳ Criar Milestone

House Keeping

- ↳ Limpar o Check Style
- ↳ Manter cobertura 100%



Não commitar  
as coisas e ter  
que refazer

- ↳ Ter cuidado ao commitar
- ↳ Estudar Git
- ↳ Não ter preguiça de commitar separado
- ↳ Fazer commits com comandos básicos do Git

Não programar  
para escrever  
documentos

- ↳ Negociar com o Presidente

Bug de deploy no  
Pétalo

- ↳ Usar mais o fórum de Pétalo

## MANTER/MELHORAR

- Comunicação (uso das ferramentas)
- Encontros informais
- Ponto no Cartaz
- Serviços no Servidor CHOREOS

## FAZER

- Maven Junina (anotar resultados)
- Limpar o Código
- 12h50 na Quarta
- Snacks na Quarta
- Email monitores p/ Nelson/Beraldo
- Tracking atrasos

Ponto:	17/06	22/06	?
LV			
Fe			
Gui			
Leo			

# Guidelines gerais (resultado de pesquisa)

- ❑ Mantenha quadros mais importantes próximos ao local da reunião diária
- ❑ Promova a alteração dos quadros por todos os integrantes da equipe
- ❑ Foque-se em necessidades específicas do momento
- ❑ Disponibilize informação fácil de ser assimilada
- ❑ Realize o preenchimento e alteração dos quadros em duplas
- ❑ Promova a praticidade e facilidade na manutenção
- ❑ Remova quadros com pouco valor

# Guidelines gerais (resultado de pesquisa)

Heuristic	Year	Experi- enced	Very Low	Low	Regular	High	Very High
# 1. Keep the most important frames/boards near to daily meeting place	2010	100%	0%	7%	7%	53%	33%
	2011	60%	0%	16.7%	41.7%	33.3%	8.3%
# 2. Divide IW related work between all team members	2010	80%	0%	3.7%	8.8%	58.8%	28.7%
	2011	65%	0%	7.7%	61.5%	30.8%	0%
# 3. Focus your IW on current specific issues	2010	90%	0%	0%	22.2%	59.3%	18.5%
	2011	65%	0%	15.4%	23.1%	38.4%	23.1%
# 4. Display information taking human brain assimilation into account	2010	100%	0%	0%	20%	37%	43%
	2011	85%	0%	5.9%	47%	35.3%	11.8%
# 5. Make your daily IW tasks done by a pair of team members	2010	90%	0%	18.5%	33.3%	40.8%	7.4%
	2011	80%	0%	18.8%	50%	25%	6.2%
# 6. Provide an easily-maintainable and practical IW	2010	100%	0%	3%	10%	50%	37%
	2011	100%	0%	5%	30%	55%	10%
# 7. Remove less valuable frames/boards from your IW	2010	87%	0%	0%	26.9%	38.5%	34.6%
	2011	95%	0%	0%	15.8%	63.2%	21%

# Guidelines gerais (resultado de pesquisa)

- Detalhamento de cada guideline:
  - ▣ Por que?
  - ▣ Como implementar?
  - ▣ Quando implementar?

<http://ime.usp.br/~renan/dissertacao.pdf>

# Ferramentas

- ❑ Jira (Atlassian)
- ❑ Mingle (ThoughtWorks)
- ❑ Pivotal Tracker (Pivotal Labs)
- ❑ Rally (Rally Software)
- ❑ Calopsita (Caelum)
- ❑ Redmine (software livre)
- ❑ Xplanner (software livre)

# Fim

▣ Dúvidas ou comentários?

Renan de Melo Oliveira  
renan@ime.usp.br

# Slides originais

Viviane dos Santos

[viviane.almeida@gmail.com](mailto:viviane.almeida@gmail.com)