

# Desenvolvimento de Software de Qualidade com Métodos Ágeis e Software Livre

*Prof. Dr. Fabio Kon*

*Centro de Competência em Software Livre  
Departamento de Ciência da Computação  
IME-USP*



- **O que é software de qualidade?**
- **Por que software livre?**
- **Por que Métodos Ágeis?**
- **Como software livre pode ajudar os métodos ágeis?**
- **Como os métodos ágeis podem ajudar o software livre?**

- **Correto**
- **Eficiente**
- **Seguro**
- **Durável**
- **de boa Usabilidade**
- **Portável**
- **Flexível**
- **Robusto**
- **de fácil Manutenção**
- **de boa Acessibilidade**
- **Belo**

## Diversas razões para adotar software livre:

- Software livre como opção ética
  - conhecimento público, disponível a toda sociedade
- Software livre e seu impacto na sociedade do futuro
  - conhecimento estratégico e fundamental demais para ficar nas mãos de apenas um grupo
- Software livre como metodologia tecnicamente interessante
  - vantagens metodológicas e técnicas

- **Software livre permite o compartilhamento de código, otimizando o uso dos recursos**
  - Menos duplicação de esforços
  - Menor custo de desenvolvimento
- **Oportunidade para melhoria da qualidade**
  - Vários olhos enxergam mais
  - Orgulho pessoal incentiva desenvolvedor a ser mais cuidadoso
  - Vários usuários envolvidos promovem melhorias e relatórios de erros
- **O mercado de desenvolvimento de software livre é um mercado local, interessante para o Brasil**

- **Percepção dos governos da Comunidade Européia:**
  - Software está ficando cada vez mais onipresente e cada vez mais importante no dia-a-dia dos cidadãos.
  - É um conhecimento importante demais para ficar fechado como segredo industrial de algumas poucas empresas: isso é ruim para a sociedade.

- **Estudos realizados na Comunidade Européia:**
  - >80% do lucro das empresas de software da CE não provém de licenças de software como produto, provém de serviços.
  - >85% dos trabalhadores de TI trabalham em empresas que mexem com software como serviço e não como produto fechado.
- Num ambiente como esse, a indústria como um todo ganhará em produtividade e qualidade se os serviços forem construídos em cima de uma plataforma de software que seja aberta e livre.

## Objetivos:

- Auxiliar o desenvolvimento tecnológico e científico do país através de 3 linhas principais:
  - Pesquisa científica e tecnológica relacionada a Software Livre
  - Desenvolvimento de Software Livre inovador
  - Disseminação de conhecimento sobre Software Livre



- Muitos métodos tradicionais:
  - Supõem que é possível prever o futuro
  - Minimizam interação com os clientes
  - Enfatizam burocracias
    - (documentos, formulários, processos, controles, etc.)
  - Avaliam progresso baseado na evolução da burocracia e não do código
- Resultado
  - Software mal escrito
    - c/ muitos erros, ilegível, de difícil manutenção

- **Indivíduos e interações** são mais importantes que processos e ferramentas.
- **Software funcionando** é mais importante do que documentação completa e detalhada.
- **Colaboração com o cliente** é mais importante do que negociação de contratos.
- **Adaptação a mudanças** é mais importante do que seguir o plano inicial.

**Nos últimos 10 anos, uma grande quantidade de ferramentas de SL foram criadas para:**

- gerenciamento de código-fonte
- gerenciamento de projetos
- testes automatizados
- integração contínua
- comunicação entre desenvolvedores
- desenvolvimento colaborativo
- planejamento de projetos

- **Eclipse / Netbeans**

- refatoração embutida
- *plugins* para automatização de *build* e de testes
- *plugins* específicos para arcabouços específicos

- **xUnit**

- SUnit, JUnit, CPPUNIT, C#Unit, PDFUnit, etc.
- disponível para praticamente qualquer linguagem

- **CVS / Subversion**

- **Wiki**

- **Selenium**
  - teste de aceitação para aplicações Web
- **XPlanner**
- **trac**
  - gerenciamento de tickets (+wiki, +integração ao rep.)
- **Maven**
  - gerenciamento de projetos Java (dependências, construção de pacotes de distribuição, documentação, portais Web, etc.)
- **CruiseControl**
  - bom quando os testes demoram para rodar













# XPlanner

[Top](#)

**Project: Janus [id=23]**

« Janus »

Melhoramentos e novidades variadas no sistema Fênix

Actions	ID	Iteration	Start Date	End Date	Days Wrk.	Stories
 	<a href="#">31</a>	Cartões Antigos	2016-03-28	2016-07-31	0,0	74
 	<a href="#">80675</a>	Cartões ainda não planejados	2011-08-01	2012-06-01	0,0	1
 	<a href="#">83084</a>	Fase 21	2009-07-15	2009-08-15	0,0	5
 	<a href="#">83535</a>	Fase 20	2009-06-15	2009-07-15	0,0	31
 	<a href="#">84327</a>	Fase 19	2009-05-15	2009-06-15	0,0	14
 	<a href="#">77519</a>	Fase 18	2009-03-20	2009-05-01	0,0	26
 	<a href="#">80565</a>	Fase 17	2008-11-30	2009-01-30	0,0	22
 	<a href="#">80566</a>	Novo Regimento	2008-08-29	2008-11-29	0,0	29
 	<a href="#">81259</a>	Fase 16	2008-10-15	2008-10-31	0,0	29
 	<a href="#">77793</a>	Fase 15	2008-05-08	2008-05-30	0,0	28

30 items found, displaying 1 to 10. [First/Prev] 1, [2](#), [3](#) [Next/Last/All]

Current Iteration

[Edit](#) | [Delete](#) | [Create Iteration](#) | [People](#) | [Export](#) | [History](#) | [Print](#)

## Iteration Fase 18 (2009-03-20 to 2009-05-01) [id=77519]

&lt;&lt; Janus &gt;&gt;

&lt;&lt; Fase 18 &gt;&gt;

Hours: Estimate 215,0, Actual 183,5, Remaining 31,5

[Save order](#)

Actions	ID	Order	User Story	I	Cust.	Progress	Act.	Rem.	Cur. Est.
	<a href="#">84747</a>	1	Incluir link "Mostrar Todos"	1	<a href="#">AAPL</a>	<div style="width: 100%; height: 10px; background-color: green;"></div>	5,0		5,0
	<a href="#">84171</a>	2	Inserir email na Lista de Orientadores	1	<a href="#">EXT</a>	<div style="width: 100%; height: 10px; background-color: green;"></div>	7,0		7,0
	<a href="#">84295</a>	3	Relatório de carga horária de docente por CPG	1	<a href="#">FK</a>	<div style="width: 100%; height: 10px; background-color: green;"></div>	7,0		7,0
	<a href="#">84167</a>	4	Erros e acertos diversos	1		<div style="width: 62.5%; height: 10px; background-color: blue;"></div>	62,5	1,0	63,5
	<a href="#">84129</a>	5	P2-ERRO: Acerto em títulos antigos	1	<a href="#">MASE</a>	<div style="width: 100%; height: 10px; background-color: green;"></div>	3,0		3,0
	<a href="#">84297</a>	6	Testes para Membros CCP - CPG	1		<div style="width: 100%; height: 10px; background-color: green;"></div>	7,0		7,0
	<a href="#">83011</a>	7	Telas de consulta: Informações da disciplina e Turmas	2	<a href="#">FK</a>	<div style="width: 50%; height: 10px; background-color: blue;"></div>	14,0	13,0	27,0
	<a href="#">80195</a>	8	Email automático s/ expiração de ementa	2	<a href="#">EXT</a>	<div style="width: 100%; height: 10px; background-color: green;"></div>	20,0		20,0
	<a href="#">77498</a>	9	Criação de nova pessoa pelo Janus	2	<a href="#">FK</a>	<div style="width: 50%; height: 10px; background-color: blue;"></div>	20,0	7,0	27,0
	<a href="#">84170</a>	10	Gerar via de conferencia do formulario de diploma	2	<a href="#">MASE</a>	<div style="width: 100%; height: 10px; background-color: green;"></div>	5,0		5,0
	<a href="#">84294</a>	11	Alteração de Histórico sem remoção de diploma	2	<a href="#">MGG</a>	<div style="width: 100%; height: 10px; background-color: green;"></div>	8,0		8,0
	<a href="#">84131</a>	12	Impressão de longas listas de presença, frequência e nota no Firefox 3.0	2	<a href="#">FK</a>	<div style="width: 100%; height: 10px; background-color: green;"></div>	8,0		8,0
	<a href="#">83351</a>	13	Controle de alunos matriculados em disciplinas (conceitos e frequências)	2	<a href="#">FK</a>	<div style="width: 25%; height: 10px; background-color: blue;"></div>	8,0	10,0	18,0

« Janus »

« Fase 17 »











« Aumento de prazo para cancelamento »

## Story: Aumento de prazo para cancelamento [id=83013]



O cancelamento de matrícula em disciplina poderá ser solicitado pelo aluno a partir do dia seguinte à data de deferimento do ministrante.

Priority:	2	Estimated Hours:	2,0 (0,0)
Customer:	nononono	Actual Hours:	2,0
Tracker:	nononono	Remaining Hours:	0,0
Last Update:	2009-05-22 14:25	Disposition:	Added
		Status:	Defined

Actions	ID	Task Name	Type	Progress	Acc.	Ori.	Est.	Act.	Rem.	Disp.	Type
   	<a href="#">83513</a>	Implementação	Feature		<a href="#">HSG</a>	2,0	2,0	2,0	0,0	Discovered	Feature
   	<a href="#">83520</a>	Produção	Feature		<a href="#">HSG</a>	0,0	0,0	0,0	0,0	Discovered	Feature



- **Git, Mercurial e Bazaar**
  - reps. distribs. integrados, facilitam *merges*, remoções, etc.
- **autotest**
  - para Ruby, testa continuamente, inteligentemente
- **testability-explorer**
  - analisa o quão difícil é testar um certo código
- **Metrics**
- **USP CCSL:**
  - JabuTi
  - Calopsita
  - XPlanner++

- Não existem muitos trabalhos sobre metodologias de desenvolvimento para Software Livre.
- As metodologias surgiram naturalmente das comunidades, indivíduos e empresas.
- a grande maioria dos projetos não usam metodologia nenhuma...
- a grande maioria dos projetos morre produzindo quase nada. (meu chute: 90%)
- a maioria esmagadora nunca atinge um grande sucesso. (meu chute: 99%)
- o 1% que sobra são as centenas de projetos de sucesso que nós conhecemos.
- como melhorar esse percentual???

- Hugo Corbucci (mestrando IME-USP, orientado por Alfredo Goldman) está estudando formas de aplicar as boas práticas de métodos ágeis em SL.
- Algumas práticas de métodos ágeis são aplicáveis em qualquer contexto e podem ser usadas em SL.
- Outras práticas necessitam de adaptações.

- **Build automático de 10 minutos**

- É sempre possível automatizar o processo de *build* para que os desenvolvedores possam montar a parte da aplicação em que eles estão trabalhando frequentemente e executá-la.

- **Integração contínua**

- Com o uso de ferramentas de controle de versão distribuídas, você nem precisa mais de uma conexão com a Internet para integrar frequentemente (inclusive, montar um "servidor" local é trivial).

- **TDD**

- Existem cada vez mais ferramentas que facilitam o desenvolvimento com TDD/BDD. Ótimas bibliotecas de teste com pequenas ferramentas de teste contínuo e excelentes ferramentas de refatoração possibilitam grande eficiência e o desenvolvimento de um sistema muito bem testado.

- **Propriedade coletiva do código:**
  - Nada mais natural em SL.
- **Padrões de formatação/estrutura de código:**
  - Muito frequente em muitos projetos mas poderia ser melhor. Padrões de código deveriam estar disponíveis na página do projeto mas também deveriam ser forçados (quando possível) no momento da integração.
- **Arquitetura simples:**
  - Qualquer projeto pode se beneficiar em ser claro e fácil de entender. Um projeto open source ganha ainda mais nisso porque existem mais pessoas capazes de dar sugestões de melhoria.

- **Uso de histórias definidas, priorizadas e estimadas de forma colaborativa.**
  - A ferramenta de gerenciamento deveria permitir que histórias sejam criados por qualquer usuário e editados de forma colaborativa (semelhante à Wikipedia?).
  - A priorização pode ser realizada sob forma de votação - usuários tem uma quantidade de votos permitidos para pedir determinadas funcionalidades.
  - Usuários "mais importantes" podem ter mais votos para representar interesses mais valiosos ao projeto (doadores ou mesmo empresas patrocinadoras).
  - A estimativa deve ser feita de forma colaborativa entre a equipe de desenvolvedores.
  - Ela pode exigir a aprovação de alguns desenvolvedores.

- **Versões (*releases*) frequentes/instântaneos:**
  - Os projetos deveriam fornecer dois tipos de versões do programa:
    1. Versões estáveis (planejadas para um período fixo à la Scrum) com congelamento das funcionalidades alguns dias antes da entrega e *release candidates*. Essas versões seriam geradas pelos administradores do projeto e ganhariam um destaque especial.
    2. Versões instáveis geradas sob demanda para os usuários. Essas versões deveriam ser geradas automaticamente com um identificador bem claro para que os *bug reports* e semelhantes sejam contextualizados.

- **Planejamento de Iteração:**
  - Só fazem sentido para as versões estáveis.
  - Não planejam o escopo. Apenas o período de tempo e um objetivo.
  - Ela serve para definir as datas para congelamento de funcionalidades e *release candidates* assim como a data de lançamento da versão estável.
  - De forma semelhante à escrita de histórias, o planejamento é colaborativo e tem data de início e de fim.
  - Durante a iteração, as histórias são escolhidas do *kanban* dinâmico e colaborativo e, quando entram em desenvolvimento, não podem mais receber votos.
  - As histórias que entram na versão são determinadas de acordo com a velocidade da equipe "estimada de acordo com as últimas 3 velocidades".



- Programação em Pares
  - *Committers* cumprem (de forma limitada) o papel do par, de forma assíncrona.
- Time Completo
- Sentar Junto
- Cliente Presente
- Papo em pé
- Trabalho Energizado

- **Software Livre é bom para o país**
- **O conhecimento aberto e compartilhado abre grandes portas para**
  - pesquisa científica-tecnológica
  - novas oportunidades para empresas consolidadas
  - oportunidades para novos empreendedores
- **Métodos Ágeis e Software Livre formam uma relação simbiótica extremamente proveitosa**
- **Todos estão convidados a colaborar e usufruir deste conhecimento**

[Ver](#) [Editar](#) [Clone](#) [Exportar](#)

## Revista especial sobre Software Livre

qua, 01/04/2009 - 17:41

O periódico científico "Information Economics and Policy", voltado para temas da área de economia em relação à informação e comunicação, lançou em dezembro de 2008 um número dedicado a aspectos empíricos do software livre.

[Leia mais](#)

## CCSL na Campus Party

seg, 30/03/2009 - 00:59



Contando com a presença de cerca de 20 alunos da graduação e da pós além de professores, o CCSL deixou sua marca no maior evento da Internet na América Latina, a Campus Party. Os membros que participaram do evento ofereceram desde palestras em áreas como desenvolvimento e software livre até um curso a respeito de grades

computacionais. O ponto alto de nossa participação foi a mobilização de grades, coordenado pelo professor Wagner Meira da UFMG, que junto com o professor Fabio Kon do IME/USP, conduziram uma palestra no palanque principal, convidando todos os presentes a cederem o seu poder computacional para processamento de proteínas no projeto Folding@home.

[Leia mais](#)

## O que é o CCSL?

O CCSL (Centro de Competência em Software Livre) do IME/USP é um centro apoiado pela FINEP, pela reitoria da USP e pelo [Projeto QualIPSo](#) e sediado no [Departamento de Ciência da Computação do IME/USP](#); contando também com colaboradores no [ICMC/USP-São Carlos](#) e [EACH/USPLeste](#).

O CCSL é membro da rede internacional [Qualipso](#) de centros de competência.

[Saiba mais...](#)

### Projetos

- AAAP
- AcMus
- Archimedes
- Borboleta
- Casamento entre Grafos
- CoGrOO (Corretor Gramatical para o OpenOffice.org)

- Colméia
- Eclipse @ IME/USP

### Calendário

« abril 2009 »

dom	seg	ter	qua	qui	sex	sab
			1	2	3	4
5	6	7	8	9	10	11
12	13	14	15	16	17	18
19	20	21	22	23	24	25
26	27	28	29	30		

### Próximas atividades

- Virtualização completa e VirtualBox

## Centro de Competência em Software Livre do IME/USP

- **Visite-nos:**

- [ccsl.ime.usp.br](http://ccsl.ime.usp.br)
- [www.agilcoop.org.br](http://www.agilcoop.org.br)

- **Escreva-nos:**

- [ccsl@ime.usp.br](mailto:ccsl@ime.usp.br)
- [agilcoop@agilcoop.org.br](mailto:agilcoop@agilcoop.org.br)

