

# Atratividade de Projetos de Software Livre

Carlos Santos Jr.

denner@ime.usp.br

12 de Novembro de 2009

Centro de Competência em Software Livre (CCSL)

<http://ccsl.ime.usp.br/>



# Contexto: Adoção e Uso de Software

- Utilização massiva e crescente de software por indivíduos e organizações (**demanda ↑**);
- Aumento significativo das opções de soluções disponíveis (**oferta ↑**):
  - Proprietários, livres, gratuitos, baixa e alta qualidade, poucas e muitas funcionalidades, etc.
- Desafio: encontrar “a melhor” solução.

**Rapid**

A fatal exception 0E has occurred at 0028:C0011E36 in UXD UMM(01) +  
00010E36. The current application will be terminated.

- \* Press any key to terminate the current application.
- \* Press CTRL+ALT+DEL again to restart your computer. You will  
lose any unsaved information in all applications.

Press any key to continue \_

# Contexto: Desenvolvimento de Software

(1/2)

- Iniciar um projeto ( $\rightarrow$  + oferta):
  - Individualmente ou em grupo;
  - Utilizando um software livre como base, ou um software proprietário como inspiração;
  - Manter o desenvolvimento internamente:
    - Custos  $\uparrow$
    - Dificuldade de melhorar o código-fonte  $\uparrow$
    - Tempo  $\uparrow$
    - Dificuldade de identificação de funcionalidades necessárias  $\uparrow$
  - Ou liberá-lo como software livre:
    - Criar uma estrutura propícia para o desenvolvimento colaborativo;
    - Atrair usuários e colaboradores, competindo no mercado de software.

# Contexto: Desenvolvimento de Software

(2/2)

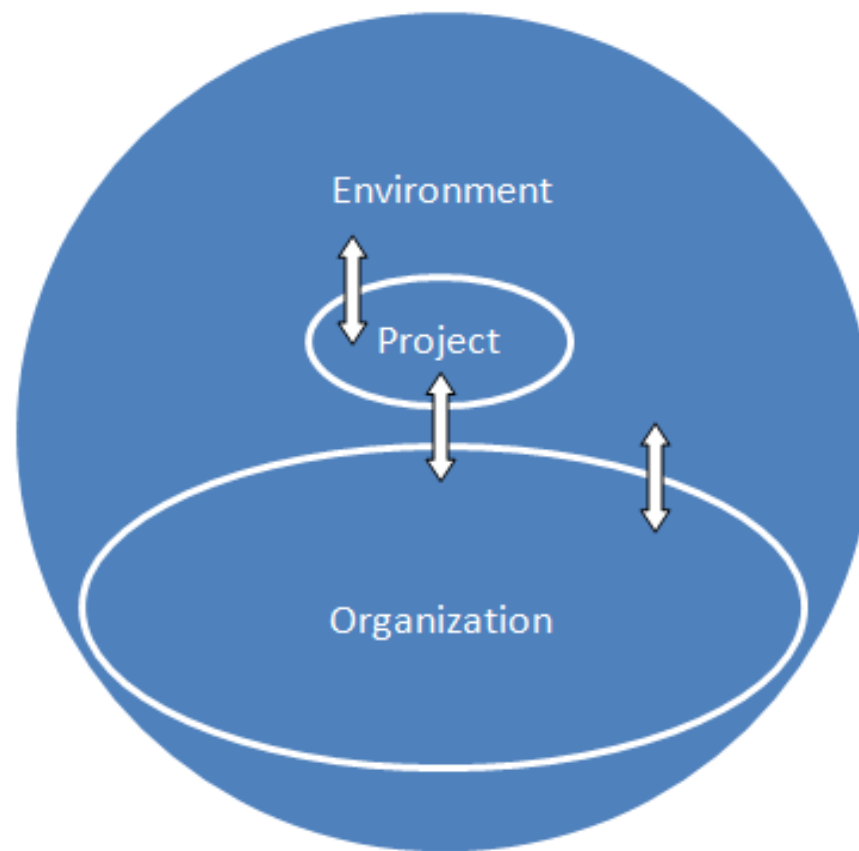
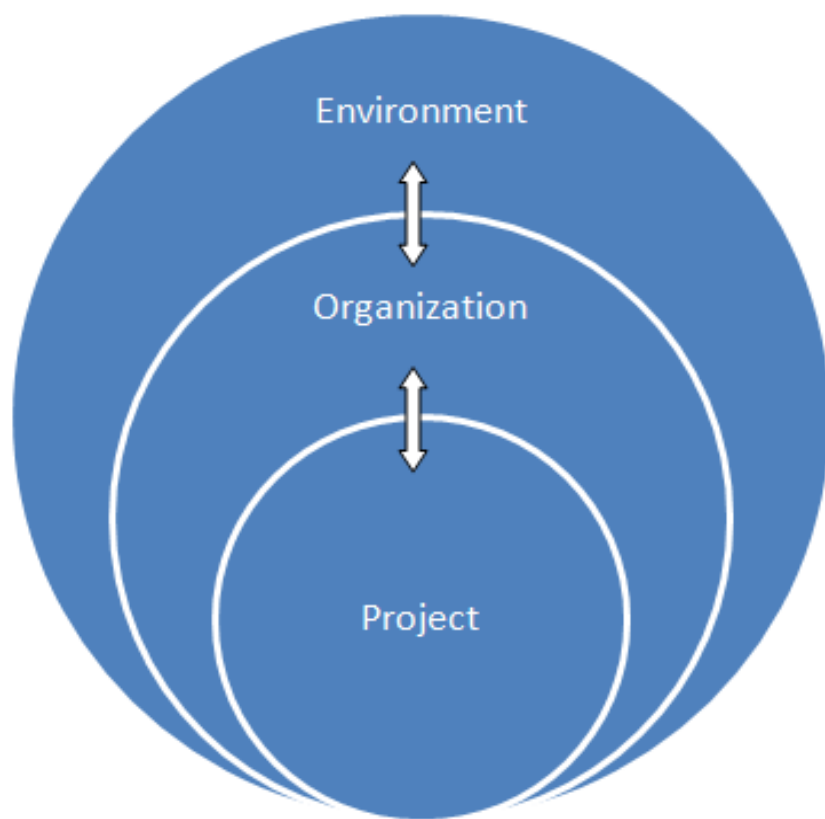
- Participar em um PSL ( $\rightarrow$  + demanda):
  - Compartilhar custos de desenvolvimento ↓
  - Dependência de fornecedores ↓
  - Dependência da comunidade do projeto ↑ (?)
- Tipos de participação em PSL, em resumo:
  - 1) Desenvolver um software internamente e “liberar” ele, esperando que contribuidores se juntem.
  - 2) Ou encontrar uma comunidade que atenda suas necessidades e fazer parte dela.

# Exemplos

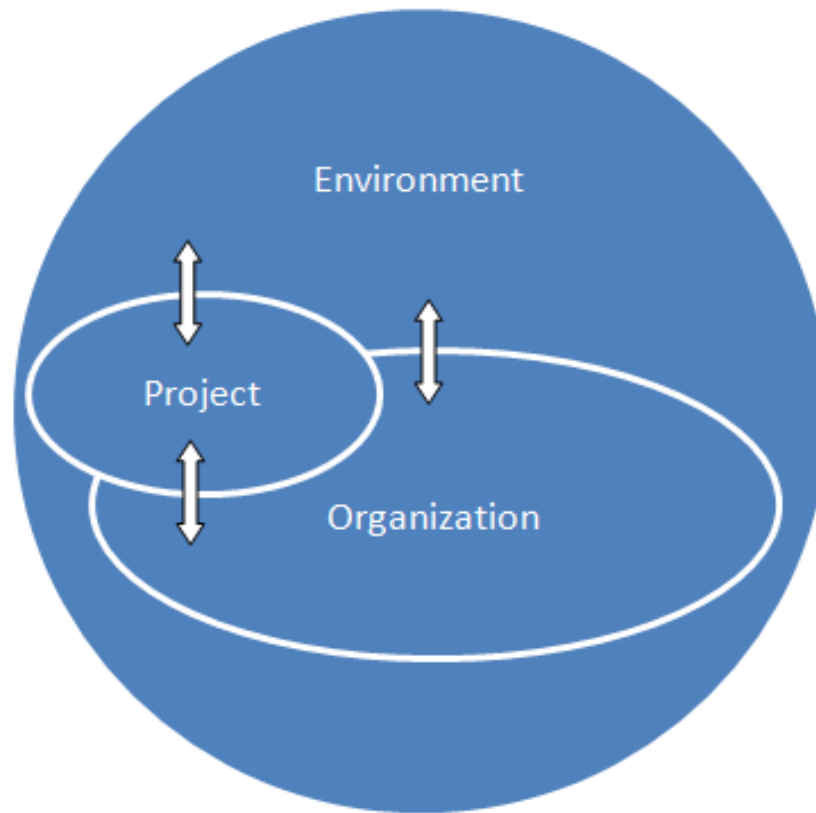
- IBM e Eclipse;
- IBM e Linux;
- SAP e Netweaver;
- Limewire;
- Mozilla e Firefox.

# Modelos de Gerenciamento de Projetos de Software

## “Make or Buy”



# Modelos de Gerenciamento de Projetos de Software “Make and Buy”





# Ambiente: Software Livre 1.0

- Metáfora do bazaar;
- Comunicação principalmente por email;
- Proporcionalmente mais voluntários;
- Licenças tipo GPL eram chamadas “virais”;
- Qualidade “em avaliação”;
- Poucos projetos;
- Repositórios inexistentes (?);
- Poucas opções de licenciamento.

# Ambiente: Software Livre 2.0

- Modelo de desenvolvimento profissional e formalizado;
- Utilização de ferramentas de gerenciamento de comunicação e desenvolvimento;
- Alto envolvimento de corporações e organizações em geral;
- Proporcionalmente menos voluntários;
- SL como parte de estratégias de mercado;
- Licenças tipo GPL são chamadas “recíprocas”.

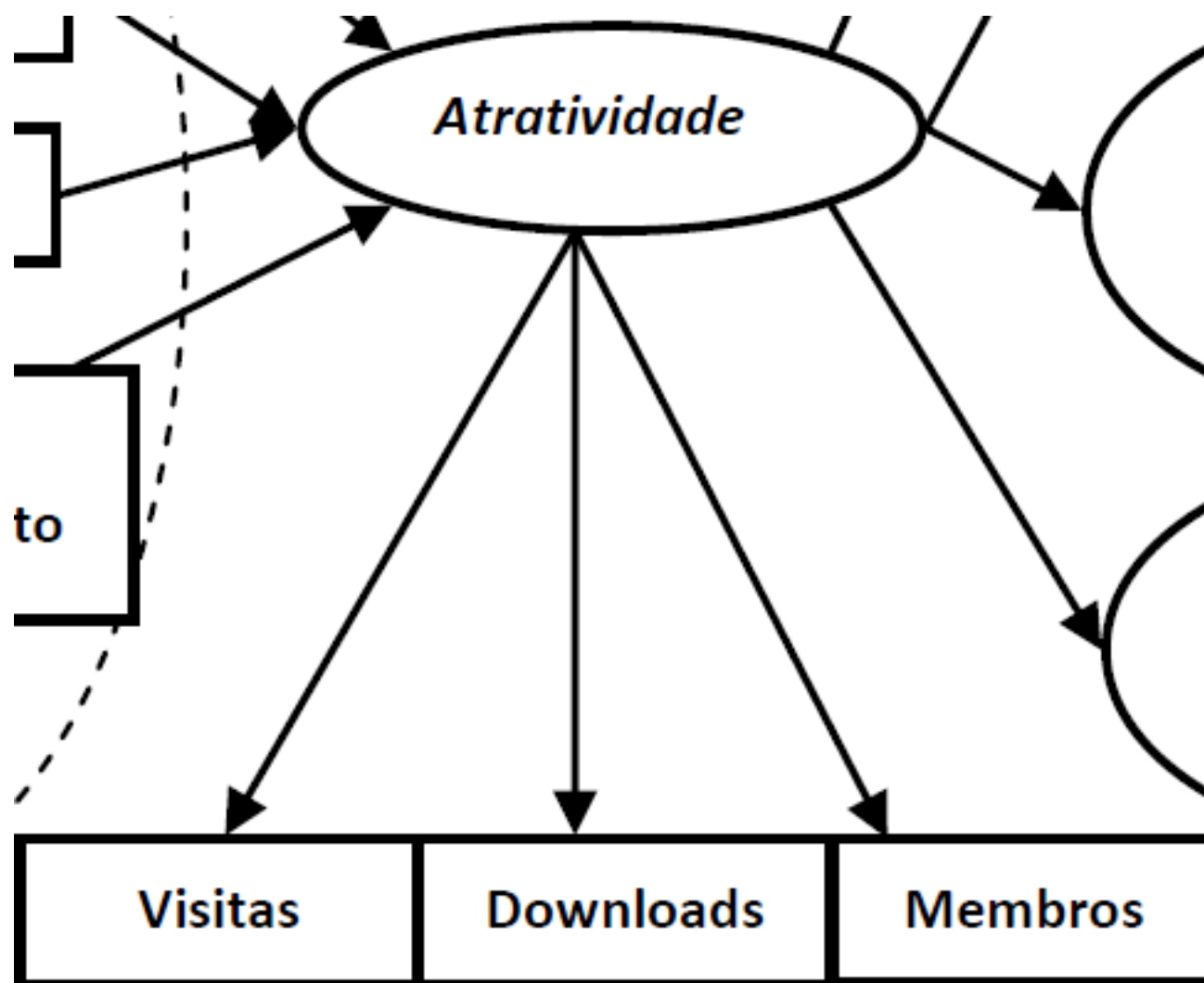
# Ambiente: Software Livre 1.0 → 2.0

- Complexidade maior:
  - Modelos híbridos, comunitários, organizacionais,...;
  - Vários repositórios;
  - Diversas ferramentas de suporte ao desenvolvimento;
  - 85+ licenças para escolher.
- Competitividade maior:
  - Milhares de projetos;
  - Quantidade de contribuidores disponíveis limitada.

# O que é Atratividade de PSL?

- Propensão ou tendência de se destacar e atender às expectativas de potenciais usuários e contribuidores;
- Operacionalmente, atratividade pode ser observada através:
  - 1) da quantidade de visitas que o sítio do projeto recebeu,
  - 2) da quantidade de vezes que o software foi baixado, e
  - 3) da quantidade de membros que o projeto possui.

# Atratividade: uma variável latente



# Por que Atratividade como Parâmetro?

- Alta atratividade leva a um número maior de bugs reportados/consertados e funcionalidades requisitadas/desenvolvidas;
  - Atratividade é impulsionadora da evolução do software e do projeto.
- Atratividade representa o sucesso competitivo do projeto.

# Utilidade do Conceito de Atratividade

- Pode ser aplicada tanto aos componentes quanto à aplicação toda;
- Permite o estudo do impacto das características do projeto em uma única variável:
  - Um projetista pode adaptar e customizar seu projeto de forma a “maximizar” a atratividade dele.
- Funciona como uma ferramenta administrativa para os criadores e mantenedores dos projetos.

# Atratividade em outros Contextos

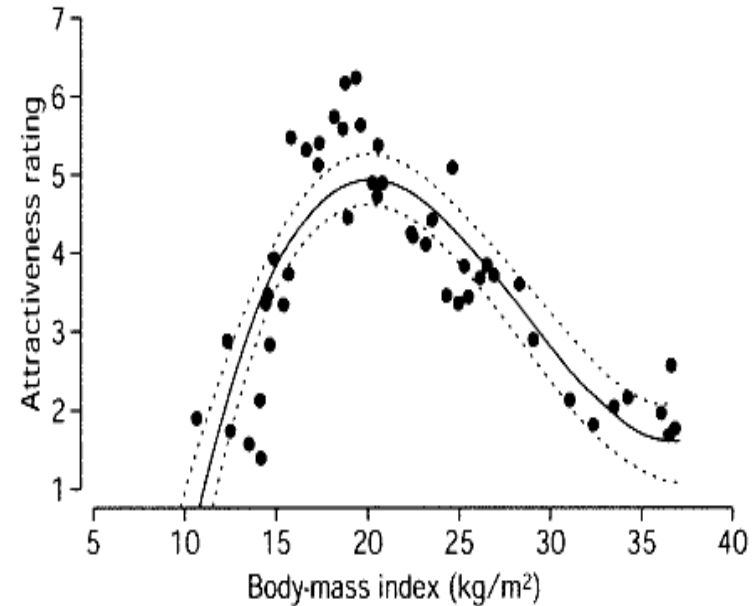
FIGURA 2 – Matriz McKinsey - Atratividade do Mercado x Posicionamento

		Atratividade da Indústria		
		Alto	Médio	Baixo
Pontos Fortes	Alto	Investimento/ Crescimento		
	Médio		Seletivo/ Ganhos	
	Baixo			Dificuldades/ Desinvestimento

Fonte: Andrade & Famá (2004)

Atratividade da Indústria: tamanho, diversificação e crescimento do mercado, estrutura competitiva, ambiente legal, relações trabalhistas, etc.

Pontos Fortes do Negócio: tamanho, crescimento, market share, posição competitiva na produção, tecnologia, etc.



Fonte: Tovee, Reinhardt, Emery, Cornelissen (1998)

Título: “Optimum body-mass index and maximum sexual attractiveness”.

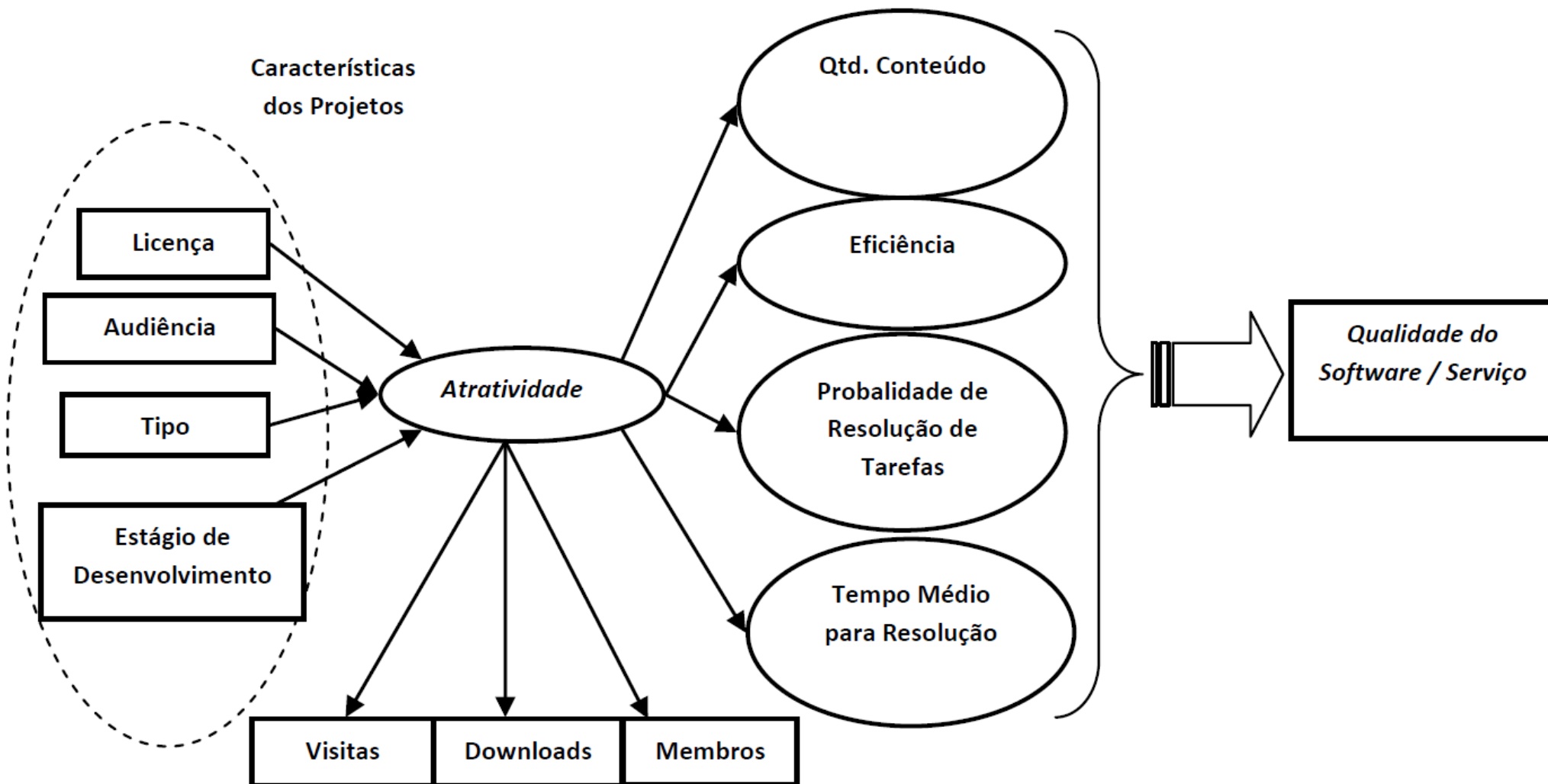
Resumo: “Young college men's preferences of women.”



# Desenvolvimento do Modelo Teórico

- O que importa aos que adotaram a estratégia “open sourcing”?
  - Menos bugs; Maior qualidade; Mais funcionalidades; Mais usuários do software; Maior visibilidade do projeto; etc.
- Sucesso do software livre (Raymond, 1999):
  - A quantidade de membros em um projeto é importante? De que forma?
  - Número de membros (“eyeballs”) e a probabilidade de localizar/resolver bugs.
  - Explica o sucesso das comunidades de software livre de acordo com o número de participantes delas.

# Modelo Teórico



# Consequências de Atratividade

- Atratividade sozinha não leva ao “sucesso”;
- Membros tem que trabalhar para melhorar o projeto:
  - Nível de atividade:
    - Quantidade de conteúdo gerado dentro do projeto (e.g., localização de “bugs”).
  - Eficiência:
    - Quantidade de “problemas” reportados e resolvidos dentro do projeto (i.e., um “bug” reportado e corrigido).
  - Probabilidade de Resolução de “Problemas”;
  - Tempo médio para completar tarefas.

# Causas de Atratividade (1/4)

- A disposição de utilizar/contribuir voluntariamente pode ser influenciada pelas restrições e permissões da licença do projeto.
- Tipos de Licença:
  - Restrição de modificação (e.g., LGPL);
  - Restrição de uso e de modificação (e.g., GPLs);
  - Restrição de uso (exemplos?);
  - Licenciamento duplo;
  - Sem restrição (e.g., MIT e BSD).

# Causas de Atratividade (2/4)

- Público-Alvo:
  - (1) usuário-final, (2) administrador de sistemas, (3) desenvolvedores, etc.
- A quantidade de membros e contribuições depende do público-alvo (Fershtman & Gandal, 2007);
- Ideia de nicho em marketing;
  - Maior o público-alvo, maior a demanda (usuários) e a oferta de contribuidores disponíveis.
- Público-alvo “representa” capacidade técnica.

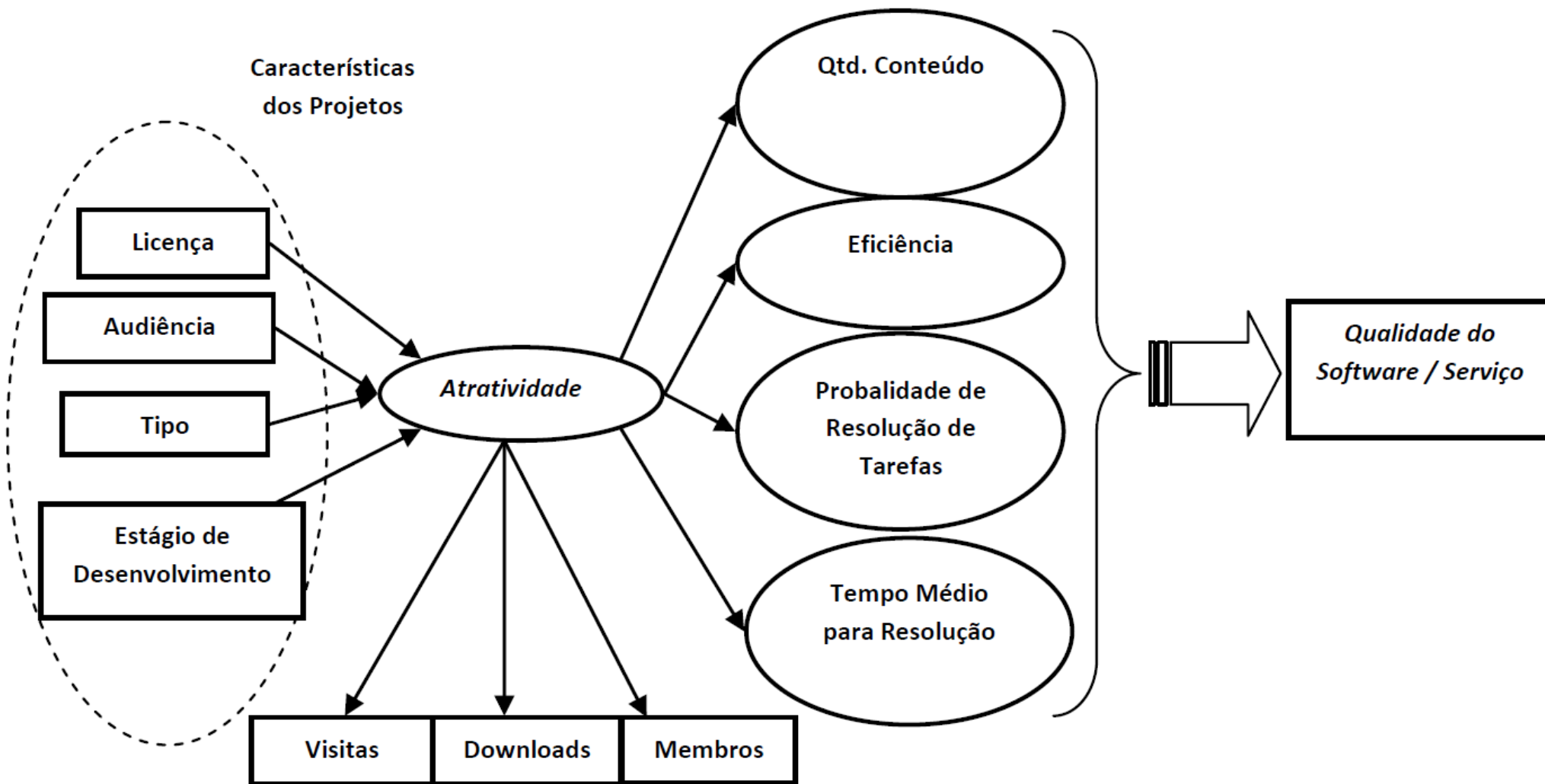
# Causas de Atratividade (3/4)

- Tipo do Projeto (domínio de aplicação);
  - Ex: genealogia, folha de pagamento, chat, jogos.
- Estabelece os “limites” de competição entre projetos;
  - Um contribuidor tende a participar de UM projeto em um determinado domínio de aplicação.
- Tipo do Projeto influencia o número de downloads e visitas, além da intensidade do trabalho da comunidade (Crowston & Scozzi, 2002).

# Causas de Atratividade (4/4)

- Estágio de desenvolvimento do software:
  - Ex: beta, estável, ou produção.
- A maturidade de um projeto de software livre influencia diretamente a probabilidade dele ser utilizado (download):
  - Funcionalidades básicas prontas.
- Da mesma forma, códigos mais maduros tendem a ser mais fáceis de serem lidos e entendidos para aprimoramento.

# Modelo Teórico



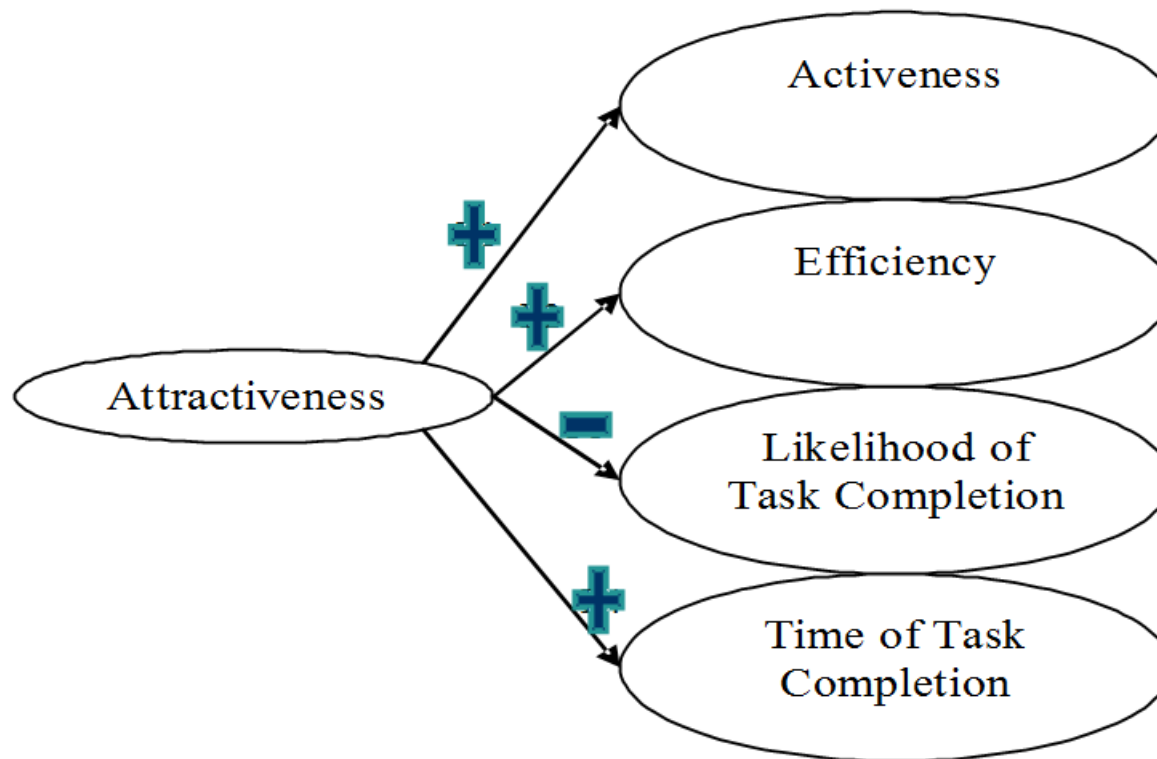


# Teste Empírico

- Fonte de dados:
  - SourceForge.net (o maior repositório)
    - Através do data warehouse da Univ. de Notre Dame;
    - Consultas SQL.
- Três coletas: Jan06, Jan07, Jan08;
- Amostras (População SF.net ~160.000):
  - 4.769, 4.611, e 4.661.
- Técnica estatística: Modelagem de Equações Estruturais.

Categorical Construct		2006	2007	2008
	<i>Sample Size</i>	4769	4611	4661
	<i>Variables' Names</i>	Frequencies		
<i>License Type</i>	no_restriction	947	1143	1327
	mod_restriction	942	1077	1197
	both_restrictions	3059	3270	3565
	dual_license	620	609	889
<i>Intended Audience</i>	end_users	2445	2548	2721
	Developers	2933	3022	3241
	system_administrators	1133	1202	1274
	others_audience	488	568	633
	advanced_end_users	155	444	616
<i>Project Type (Topic)</i>	communications	773	181	220
	Database	392	198	228
	Desktop	235	126	139
	Education	171	148	180
	Games	606	219	254
	Internet	1184	241	274
	Multimedia	714	38	50
	Office	328	137	149
	Other	138	149	171
	Printing	39	43	45
	Religion	21	22	22
	Scientific	567	117	130
	Security	156	126	146
	Sociology	28	15	17
	Software	1158	572	604
	System	923	45	50
	Terminals	40	11	14
text_editor	215	51	57	
<i>Life-cycle Stage</i>	Planning	243	638	983
	pre_alpha	318	544	692
	Alpha	818	985	1117
	Beta	1811	1898	2009
	Production	2251	2299	2462
	Mature	208	227	243
	Inactive	76	111	154

# Principais Resultados (1/2)



# Principais Resultados (2/2)

- Essas características influenciam atratividade positivamente:
  - Licença (licenciamento duplo, restrição de uso e modificação - GPLs);
  - Público-alvo (usuário-final, desenvolvedores);
  - Tipo do Projeto (impressão, segurança);
  - Estágio de Desenvolvimento (beta, produção, maduro).
- Essas características influenciam atratividade negativamente:
  - Licença (sem restrição);
  - Público-alvo (outros);
  - Tipo do Projeto (banco de dados, outro);
  - Estágio de Desenvolvimento (planejamento, pre-alpha, alpha, inativo).

# Considerações Pontuais (1/2)

- A categoria “outros” deve ser evitada;
- Priorize a liberação de software útil e entendível, com características de programas mais maduros;
- Tempo de existência (idade do projeto) não influencia quantidade de conteúdo gerado nem eficiência (atratividade sim!);
- Os efeitos das restrições das licenças parecem ser amplos e ainda não são completamente conhecidos (escolha com cuidado!);

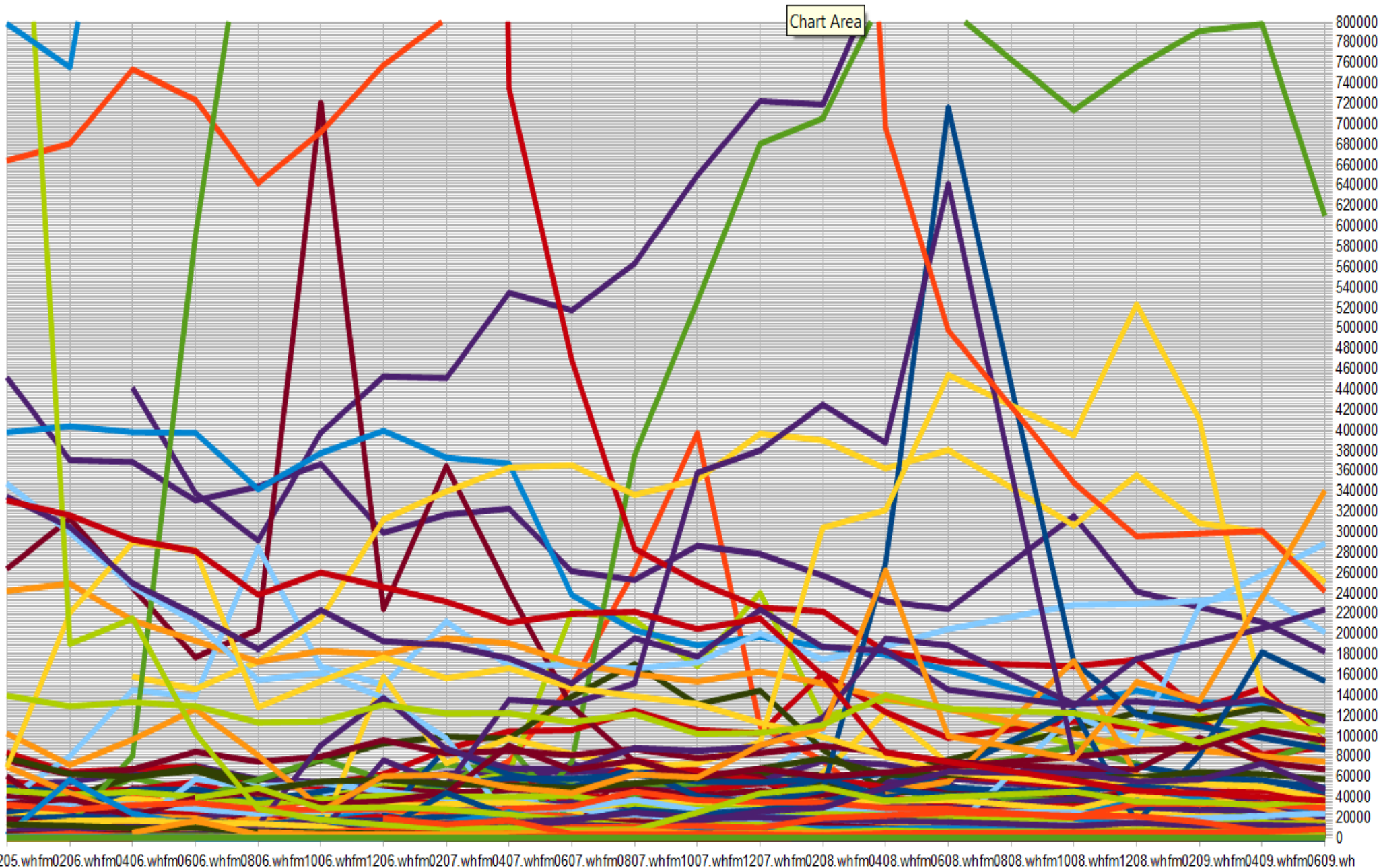
# Considerações Pontuais (2/2)

- Não existe relação causal entre visitas, downloads e número de membros, mas sim uma causa em comum;
- Qtd. de conteúdo aumenta com o número de desenvolvedores;
  - Probabilidade de inovação ↑;
  - Qualidade ↑;
  - Eficiência ↑;
  - Tempo de resolução de tarefas ↑.
- Necessidade de administrar a atratividade do projeto, aumentando sua visibilidade (melhor posicionamento);
- O estudo das características dos projetos e dos impactos delas viabilizam planejar a criação/adaptação de projetos de software livre;
  - Foco: desenvolvimento de um ambiente mais propício para contribuições ocorrerem.

# Direções Futuras

- Atratividade do Projeto e Características do Código:
  - Tamanho;
  - Acoplamento;
  - Ausência de coesão em métodos; etc.
- Sustentabilidade;
  - Taxa de crescimento de atratividade;
  - Projetos, em geral, de fato encontram novos contribuidores?

# Webhits



fm1205.whfm0206.whfm0406.whfm0606.whfm0806.whfm1006.whfm1206.whfm0207.whfm0407.whfm0607.whfm0807.whfm1007.whfm1207.whfm0208.whfm0408.whfm0608.whfm0808.whfm1008.whfm1208.whfm0209.whfm0409.whfm0609.wh