

# Modelo de Referência para Desenvolvimento de Software em Ambiente Acadêmico

## 1 Pergunta de Tese

Qual processo tem sido cumprido para o desenvolvimento de projetos acadêmicos e como melhorá-lo?  
ou: Como melhorar o processo cumprido para o desenvolvimento de projeto acadêmico?

Estratégia: Quem são os interessados em processos academicos e quais sao seus interesses?  
alunos– manutenção de projetos, orientadores...

## 2 Objetivos

O que queremos atingir com o uso do processo?

- Melhorar a manutenibilidade das aplicações (evolução de projetos academicos): O que preciso para alcançar manutenibilidade (fazer revisao bibliografica – alguns resultados estao na coveb) Um elemento que se destaca é o uso de processos e de documentação. Isso envolve entao projetar um novo processo, que atenda as necessidades do contexto que estou considerando (ambiente academico). Estou fazendo um levantamento das necessidades dos clientes do meu processo. Os clientes que estou considerando são: estudantes, coordenadores, empresas, agências de financiamento. Os processos estão sendo identificados a partir dos questionarios e de revisao da literatura. Considerar que os processos sao tipicamente distribuidos e colaborativos. Qual o impacto disso no meu processo? estudar estes dois casos. O que mais interfere na manutenção e que eu preciso considerar? qual o tipo de manutenção cada item do meu processo ataca?
- Estou identificando processos a partir de revisão da literatura (o que os autores desejam), descrição de projetos acadêmicos (o que é feito na prática) (artigos e repositórios de projetos), estudos de projetos reais, questionários, modelos de referência, entrevistas com as pessoas. Pra obter o conhecimento estou usando a técnica de aquisição de conhecimento do mestrado da Andrea.
- Transferência de conhecimento e transferencia de tecnologia: captura de conhecimento, aprendizado por exemplo, design rationale. É difícil implementar a parceria universidade-empresa para desenvolvimento de software por muitos motivos (14; 16; 18). Neste trabalho, estamos oferecendo uma pequena contruibuição neste sentido: estamos pensando na trasferencia tecnologica. É importante mencionar que tivemos uma experiência (SAFE) que nos possibilitou

tirar algumas conclusões a respeito da parceria. (fornecer os insights). Os artigos referenciados falam de atividades para a parceria entre universidade/industria.

- Reuso entra?

Fatores de qualidade que estou considerando na definição do processo: simplicidade, flexibilidade

Foco: projetos pequenos, com escopo bem delimitado, desenvolvido por poucas pessoas (porque não conseguiria avaliar em projetos grandes).

Um exemplo interessante que posso utilizar: curso de HCI, Graça, coteia. Oferece passos simples, templates

Como é o treinamento do meu processo?

Sugerir estratégia de implantação

### 3 Projeto de processo

1)- Davenport ressalta que um processo deve ser projetado de modo a produzir resultados que satisfaçam às exigências do cliente. O projeto de um processo deve ter um bom entendimento de quem são os clientes do processo e o que eles querem dele.

- Clientes do processo academico

- estudantes; esperam documentação dos projetos (questionario sobre manutenção),
- coordenadores de projetos; esperam gerenciamento dos projetos (questionario com professores)
- empresas; esperam treinamento e capacitação, transferencia de conhecimento e de tecnologia
- Instituições financiadoras, desenvolvimento de projetos de pesquisa, prestação de contas, gerenciamento de recursos, gerenciamento de recursos financeiros esperam relatorio de prestação de contas, relatorios científicos, teses, dissertações.

outros clientes:

- administrativos – não estou considerando
- Instituições de ensino ou de pesquisa, esperam desenvolvimento de projetos de pesquisa e cumprimento de prazos. Esperam receber relatorios científicos e de atividades, teses, dissertações
- candidatos a participação de grupos de pesquisa (alunos de pós graduação) – esperam um cronograma com datas de submissao de documentos, resultados, matriculas. Isso envolve todo o processo de seleção de alunos
- secretários, esperam um processo de aquisição de software, equipamentos, materiais
- avaliadores do projeto, que podem ser os membros de uma banca ou avaliadores dos orgaos de pesquisa.

As organizações precisam definir um responsável pelo funcionamento de cada processo como um todo (process owner).

2)- Savi e Franzoni indicaram que para o modelo de referencia ser construido foi necessario saber, antes de tudo, quais os processos normalmente existem na incubadora (no meu caso, no ambiente academico)

Para a elaboração do modelo de referencia proposto, foram utilizadas as fontes descritas anteriormente. Em relação aos questionarios, foram considerados todos os processos que foram assinalados como importantes por mais de 50% das pessoas que responderam. Os resultados obtidos são apresentados a seguir.

#### 1. **Cliente:** Estudantes

- Necessidades (o que esperam): Documentação, comunicação (questão 9), garantia de qualidade de produto e de processo (questão 11)
- Fonte: Questionário 1, aplicado a alunos (manutenção de projetos academicos). Questões 6, 9, 10 e 11
- Processos: Processos de ciclo de vida (prototipação e metodos ageis), controle de versões, controle de mudanças, atividades de verificação do software, atividades de validação do software
- Produtos: (1) Artefatos resultantes do ciclo de vida: plano de projeto da pesquisa (contendo definição de escopo, objetivos, restrições), documento de requisitos, diagrama de casos de uso, diagrama de classes, código fonte (questões 6 e 10); (2) versões do sistema; (3) registro de mudanças, resultados da verificação do software, resultados da validação do software (questão 11).

#### 2. **Cliente:** Coordenadores (orientadores)

- Necessidades (o que esperam): qualidade do produto, qualidade do processo, transferência de conhecimento, continuidade e evolução dos projetos, qualidade científica (questão 5, incluindo a parte de comentários)
- Fonte: Questionário 2, para coordenadores (Questão 5)
- Processos: Processos de ciclo de vida, atividades de validação do software, controle de prazos, atividades para transferência de conhecimento, atividades que auxiliem a definição rigorosa da proposta, definição, formalização de procedimentos e métodos (definição de metodologia científica), treinamento
- Produtos: pelo questionario nao sei quais artefatos

A seguir são apresentadas as necessidades dos clientes do processo acadêmico de acordo com os resultados obtidos a partir de revisão da literatura

#### 1. **Cliente:** Estudantes

#### 2. **Cliente:** Coordenadores

- Necessidades (o que esperam): atividades de gerenciamento de projetos, atividades de pré-desenvolvimento, integração do sistema (25)
- Processos
  - definição e gerenciamento de tarefas, definição e gerenciamento de cronograma, análise de requisitos e treinamento no domínio da aplicação, projeto e codificação de forma relacionada, validação de sistemas, testes, documentação e revisão técnica (25)

- gerenciamento de projetos, gerenciamento de conhecimento, documentação, registro de bugs, comunicação, controle de versões, controle de erros, projeto e geração de código, testes, peer-review, garantir que outras pessoas se tornem familiar com os projetos desenvolvidos (2)
- Prototipação, suporte a distribuição e colaboração, métodos ágeis, testes, verificação, comunicação, brainstormings, gerenciamento de conhecimento (5)

### 3. **Cliente:** Empresas

- Necessidades (o que esperam): (1) transferência de conhecimento (boas práticas de Eng. de Software) (14; 26); (2) parcerias com universidades (16)
- Processos:
  - Elaboração de material para treinamento, Seleção de parceiro, negociação de suporte financeiro para universidade e bolsa de estudos para estudantes envolvidos, identificação de períodos em que o estudante possa fazer intercâmbio na indústria, definição de intercâmbio de profissionais e professores na empresa e na universidade, criação de infra-estrutura na universidade de forma a suportar o desenvolvimento de software no principal domínio de aplicação da indústria (incluindo equipamentos, documentação e configuração), realização de pesquisa em engenharia de software em áreas de interesse mútuo, transferência de solução e tecnologia para o parceiro da indústria, elaboração de plano de operação (detalhes do projeto de pesquisa, métodos, pessoas, técnicas e artefatos a serem produzidos), definição de papéis e responsabilidades, submissão de proposta para agência financiadora indicando direções de pesquisa, recursos financeiros requisitados e resultados esperados, avaliação anual dos parceiros envolvidos em relação aos resultados obtidos (parcerias programadas para um ano, podendo haver renovação), captura e reuso de conhecimento, uso de modelos para desenvolvimento, técnicas de verificação, validação e testes (16)
  - definição de objetivos claros (18)
  - identificar o que cada participante ganha (win-win situation), relacionamento entre pesquisadores e profissionais, o principal pesquisador deve conhecer a pesquisa e o desenvolvimento da empresa, o membro da indústria responsável deve ter visão de pesquisa, contato pessoal e interações frequentes entre os membros (intercâmbios), definição de papéis e responsabilidades, parcerias programadas para um ano podendo haver renovação, planejamento de ferramentas e equipamentos a serem utilizados, definição de cronograma, milestones e benchmarks e tarefas, organização de reuniões para gerenciamento de projeto, estabelecer acordo sobre confidencialidade de informações da indústria e devolução de documentos após o desenvolvimento do projeto, seleção de informações que podem ser disponibilizadas (pela indústria), negociação de suporte financeiro, identificação de recursos fornecidos por cada uma das partes envolvidas, geração de relatórios e documentos pelo pesquisador responsável sobre o funcionamento de laboratórios, relações entre academia e indústria, publicações e contratos incluindo direitos de propriedade do software (copyrights e patentes), divisão de lucros, controle da especificação dos projetos e direito de publicação de informações e resultados obtidos; comunicação entre os participantes, gerenciamento de projetos, controle de troca de informação entre países diferentes (4)

- Produtos: material didático, relatórios técnicos, contratos

Processos relacionados aos requisitos de processos acadêmicos: distribuído, colaborativo (processos separados ou pulverizados nos outros processos?)

Outra forma de identificar os processos que serão incluídos no modelo de referencia é observando as atividades que são comumente cumpridas em projetos de pesquisa que são desenvolvidos. Duas abordagens serão usadas: observar a descrição das atividades cumpridas em artigos científicos e observar as atividades cumpridas diretamente em projetos (observar a documentação, entrevistar pessoas).

A seguir são apresentados os resultados da análise de práticas de engenharia de software registradas em artigos científicos. Critérios para seleção dos projetos analisados: artigos/documentos que explicitavam o caráter acadêmico dos projetos, artigos completos (resumos e artigos com menos de x páginas foram evitados, por fornecerem apenas uma descrição breve de atividades de processo). Projetos foram buscados em periódicos, anais, teses, dissertações, repositórios de projetos na internet. Algumas pessoas foram contactadas e foram solicitados artigos e relatórios que pudessem descrever processos de projetos desenvolvidos por elas.

1. **Projeto Genesis:** definição de tema, gerenciamento de conhecimento, análise de requisitos, liberação de releases constantemente (release early, release often), reuso, controle de versões, decisões de plataforma, linguagens e infra estrutura em fases iniciais, divulgação do projeto (planejar e cumprir as tarefas para divulgar) por exemplo por meio de artigos, apresentações em conferências, CDs, sites, feiras, sourceforge, suporte a comunicação, gerenciamento de riscos, gerenciamento de cronograma, gerenciamento de configurações, testes, revisões técnicas, introdução de elementos de XP, promover a colaboração, promover o trabalho distribuído ( ? )
2. **Projeto COSPA:** análise de requisitos (questionários, entrevistas, histórias dos usuários), desenvolvimento de protótipo, análise custo/benefício da aplicação, construção e disponibilização de repositório de conhecimento, divulgação de resultados, (31)
3. **Projeto MVCASE** (Daniel Lucrédio): especificação do comportamento das operações, melhoria de usabilidade, alguma documentação, registro de bugs e sugestões, análise do pedido de alteração, gerenciamento de pessoas, gerenciamento de tempo, modelo de classes, gerenciamento/atualização de home page, arquitetura do projeto, implementação, testes para verificar a presença de erros, gerenciamento de mudanças (descrição, implementação e disponibilização da mudança), publicação e disponibilização das mudanças. Descrição da mudança envolve descrever o problema que originou a mudança, mudanças que foram executadas, os arquivos modificados, quem registrou a mudança, suas informações de contato, data em que a mudança foi registrada. Os autores identificaram um processo necessário para o desenvolvimento da MVCASE, que deveria: ser implantável em uma instituição acadêmica, permitir controlar mudanças, envolver um número suficiente de desenvolvedores e pesquisadores, dirigido ao mercado, considerações inovações propostas por pesquisadores na academia.
4. **Projeto CARA** (Computer Assisted Resuscitation Algorithm) (11; 20): uso de um processo incremental, especificação formal (usando linguagens para especificação de sistemas de tempo real), análise de requisitos (documento de requisitos escrito usando linguagem natural), projeto arquitetural, implementação, testes (plano de testes, casos de testes, execução), avaliação de

usabilidade, modelagem de sistemas de tempo real, técnicas de geração de código automático, revisão do sistema, projeto de dados, controle de versões, inspeção de requisitos, registro de design rationale, treinamento em técnicas de especificação formal, definição do escopo do projeto, gerenciamento de tempo, orçamento e pessoas, descrição de estados de máquina de estados finitos, verificação, inspeções (provas formais)

5. CAOS - Collaborative Application for Open Storage (repositorio Java.net, <https://caos.dev.java.net/source/browse>)  
Processos comerciais (pedidos de propostas), processos administrativos (gerenciamento de pessoas, melhoria de processos), processos de desenvolvimento de software: Análise de negócios (contactar o cliente, fornecer informação de status de projeto, responder às expectativas dos clientes), Análise de software (analisar, documentar e manter use-cases, requisitos, arquitetura e modelo de base de dados), teste de software (definir plano de testes, incluindo casos de teste e critérios de aceitação, implementar, executar e validar casos de teste, fornecer relatório de teste), Gerenciamento de projeto (revisão do status do projeto, revisão das estimativas e da documentação de projeto, gerar plano de projeto, registrar status do projeto), Gerenciamento (identificar prioridades, resolver questões estratégicas, oferecer suporte ao desenvolvimento do projeto), Processos de qualidade (treinamento em garantia de qualidade, garantia de qualidade de software, verificação do produto, manter configuração do projeto, coleta e registro de métricas, garantia e melhoria do processo, elaborar plano de qualidade, avaliar o projeto do software em relação ao plano do projeto, planejar e gerenciar configuração do projeto, configurar e manter o ambiente de gerenciamento do projeto, estabelecer, criar e manter baselines, gerar e publicar releases, identificar objetivos estratégicos para o processo, suportar a definição do processo de qualidade, definir plano de ação para desvios do projeto, fornecer recursos para a realização de atividades do processo de qualidade, registrar pedidos de mudanças, definir responsabilidades para pedidos de mudanças, homologar mudanças) (informações apresentadas em artigo submetido ao SIMS2005, na pasta RelatodeExperiencia)
6. ADVS (Lawyer activities control system)(<http://advs.tigris.org/>): análise de requisitos (casos de uso), definição de casos de testes, definição de procedimentos de testes, planejamento da gerência de configuração de software (organização, responsabilidades, interfaces, ambientes, ferramentas), gerência de configuração (Identificação de Configuração, Controle de Mudança e Configuração, controle de versões), treinamento, plano de desenvolvimento do projeto, gerenciamento de pessoas, gerenciamento do projeto, gerenciamento de riscos (Principais riscos ao projeto, Procedimentos de análise dos riscos), gerenciamento de custos e de prazo, descrições dos envolvidos e usuários (Resumo dos Usuários, Ambiente dos Usuários, Resumo das Principais Necessidades dos Envolvidos ou Usuários Alternativas e Concorrência)
7. Projeto Hyperdev: análise de requisitos (identificação de requisitos gerais, requisitos funcionais, requisitos da interface com o usuário), análise e projeto (diagrama de classes, projeto arquitetural, modelagem da base de dados, definição de módulos e funcionalidades associadas a cada um) desenvolvimento de protótipos evolutivos, registro de decisões de projeto (por meio de arquivo de mensagens eletrônicas e descrições das deliberações correspondentes a cada versão do sistema e documentação de idéias para futuras versões), documentação dos módulos (por meio da descrição da implementação. Corresponde a um arquivo HTML que contém ligações hipertextuais a programas que executam os módulos), gerenciamento de atividades (coordenação das atividades, determinação da prioridade das tarefas e planejamento das atividades da versão seguinte do sistema), elaboração do manual de uso do sistema. (19)

8. Projeto Canto Livre: definição de escopo, análise de requisitos (documento de requisitos e casos de uso), análise do projeto, projeto arquitetural, testes planejamento de atividades (gerou um cronograma de atividades) e acompanhamento do progresso do projeto (por meio de reuniões e uso da ferramenta Atlassian, que tornou mais efetivo o acompanhamento da equipe distribuída), revisões, planejamento do projeto, elaboração de propostas técnicas e comerciais, controle de mudanças e gerenciamento das atividades usando o Issue Tracker do site do Tigris, controle de versões, estimativa de esforço, Uso de ferramentas para trocas de mensagens instantaneas e emails, reuniões presenciais, uso de ferramenta de colaboração via web para atualizar as informações do projeto, em que havia duas visoes, uma para o processo e outra para o projeto. Outras ferramentas utilizadas: Atlassian Confluence (ferramenta de colaboração), Issue Tracker (controle de atividades a serem cumpridas), ArgoUML, Eclipse, WinCVS (Brito et al.)
9. Projeto Tupan: projeto participativo foi priorizado, avaliação dos processos e produtos de trabalho, auditoria de processo, gerenciamento do repositório de versoes, análise de requisitos (documento de requisitos e casos de uso), análise do projeto, projeto arquitetural, testes, acompanhamento do progresso do projeto, revisões (questionamentos, dicas e duvidas dos revisores eram registradas no proprio documento sendo revisado), planejamento do projeto, controle de mudanças e gerenciamento das atividades usando o Issue Tracker do site do Tigris, controle de versões, estimativa de esforço, Uso de ferramentas para trocas de mensagens instantaneas e emails, reuniões presenciais, uso de ferramenta de colaboração via web para atualizar as informações do projeto, em que havia duas visoes, uma para o processo e outra para o projeto. Outras ferramentas utilizadas: Atlassian Confluence (ferramenta de colaboração), Issue Tracker (controle de atividades a serem cumpridas), ArgoUML, Eclipse, WinCVS (30)
10. Protótipo do modelo APSEE: análise de requisitos, análise (diagrama de pacotes, diagrama de classes, transição de estados,) projeto arquitetural, identificação de componentes, construção de prototipo, estudos de caso (23)
11. Prosoft cooperativo: especificações formais, descrição formal de operações que o sistema executa, especificação dinâmica dos processos que compõem o editor (24)
12. Controle de acesso para sistemas distribuídos Java, (<https://ablp.dev.java.net/>): definição de escopo, análise de requisitos, projeto arquitetural, desenvolvimento de protótipo (29)
13. Projeto Balehar, <http://balehar.tigris.org/>: análise de requisitos, registro de design rationale (troca de mensagens), diagrama de classes, modelo de base de dados, manual de usuário, implementação (projeto desenvolvido na França)
14. Projeto Codecrawler: análise de requisitos, registro de lições aprendidas, projeto arquitetural (justifica a falta de documentação mencionando o ambiente acadêmico de desenvolvimento)
15. Development of a design history information system: análise de requisitos, tradução dos requisitos e dos resultados de pesquisa em um conjunto de especificações (modelos detalhando os requisitos), pesquisa na literatura sobre sistemas PDM (Product Data Management) para obter background para implementação, projeto arquitetural, modelo de dados, implementação e instalação do sistema, avaliação (estudos de caso)
16. Seurat: análise de requisitos, projeto arquitetural, implementação, avaliação
17. Projeto Oscar: análise de requisitos, projeto arquitetural, controle de versões, teste de unidade, revisão de código, aplicação de práticas de XP,

18. Adelfe (relatório tecnico): análise de requisitos, modelagem conceitual
19. Integração de Conhecimento em um Ambiente de Desenvolvimento de Software (AssistPro): análise de requisitos, descrição de cenários de utilização, descrição textual do universo de discurso, modelo de tarefas, modelagem arquitetural, prototipação (Falbo)
20. Arquitetura para Gerenciamento de Conhecimentos explícitos sobre Processo de Desenvolvimento de Produto: Análise de requisitos, projeto arquitetural, prototipação, modelagem de dados, modelagem de interface, descrição de cenário de aplicação, etapas para implantação da solução, avaliação (1)
21. AutorE: análise de requisitos, modelagem conceitual, projeto arquitetural, prototipação (27)
22. Projeto Mae, universidade da california: análise de requisitos, projeto arquitetural, prototipação
23. Projeto Hipikat: análise de requisitos, projeto arquitetural, prototipação
24. Projeto PDE (desenvolvido no contexto do projeto E-WEBPROJECT): definição de requisitos funcionais e não funcionais, diagrama de casos de uso, diagrama de atividades, diagrama de sequencia, modelo de classes, diagrama de estados, modelo da interação entre objetos, projeto arquitetural, implementação, diagrama de colaboração (15; jun)
25. Desenvolvimento do ambiente ABCDE-Feature: análise de requisitos, análise (diagrama de classes), projeto arquitetural, implementação (21)
26. Projeto PERF: análise de requisitos, projeto arquitetural, implementação (8; 10)
27. Especificação e desenvolvimento da ferramenta C&L: controle de versões, análise de requisitos, gerenciamento de tarefas e de pessoas, registro de dificuldades, dúvidas, soluções e resultados parciais em lista de discussão, definição e tratamento de cenários (mapa de relacionamentos entre cenários), implementação, elaboração de casos de testes, execução de testes, inspeções de cenários baseadas em checklist, acompanhamento da correção de defeitos, verificação de cenários mantidos, avaliação por meio de estudo de caso (28)
28. Projeto Adequas: análise de requisitos, projeto arquitetural, avaliação por meio de estudo de caso (9; 22).
29. Projeto ProGrid: análise de requisitos, projeto arquitetural, implementação (32).
30. NSCP (distributed system for the design of a network and service control platform): análise de requisitos, projeto arquitetural, projeto de interface, implementação (13) – projeto desenvolvido na University of Ottawa, Canada.
31. Algorithm for Establishing LSPs for Multicast Communication: análise de requisitos, projeto arquitetural, implementação, validação (6)
32. Projeto WQoS: análise de requisitos, análise comportamental (uso da notação UCM), implementação, avaliação por meio de estudos de caso (7)
33. Projeto Strathcona: definição de cenário, implementação, avaliação (12). Desenvolvido nos Estados Unidos



34. GQMTTool: definição de requisitos, diagrama de classes, projeto arquitetural, implementação, validação (17) - desenvolvido na Italia
35. Application for the Critical work of Email and Information Management: gerenciamento de pessoas e definição de papéis, análise de requisitos (documento de requisitos, registro de estórias), estimativa de tempo para desenvolvimento das estórias, revisões técnicas, planning game, implementação, testes de aceitação (3) desenvolvido nos Estados Unidos  
ver algum projeto do fabio kon
36. ProjectSpace:

A seguir são apresentados os resultados da análise de projetos para os quais uma observação mais detalhada dos artefatos foi realizada.

### **3.1 Projeto SAFE – Software Engineering Available for Everyone**

### **3.2 Projeto Regra**

### **3.3 Wiki-RE**

### **3.4 No-risk Planning**

### **3.5 Projeto DocRationale**

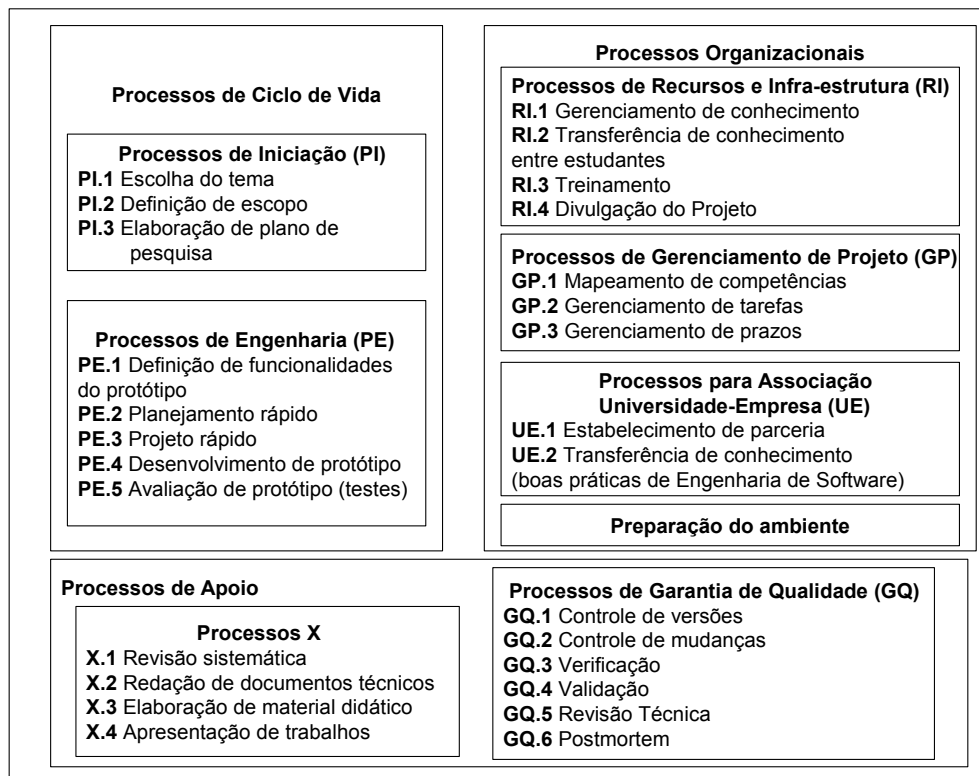
## **4 Modelo de Referencia**

A Figura 1 apresenta uma visao geral do modelo de referencia definido

Além dos elementos da 15504, posso colocar comentários – DR de processo, sugerindo abordagens, por exemplo, XP Research.

#### **Observações:**

- Elaboração de plano de pesquisa: envolve as atividades de definição de tema que eu havia colocado no modelo inicial, incluindo definição de objetivos, motivacao, decisões de plataforma.
- Boldyref comenta que projetos de open source fazem gerenciamento para evitar os mesmos problemas que ocorrem em amb. academico. Olhar o que posso pegar de open source para o meu trabalho.
- Ambati, Boldyref, Paula Filho e autores do easyprocess me fornecem requisitos para um processo academico: suporte a colaboração (promover a comunicação), suporte a trabalho distribuido (sugere metodos de open source) (2? );
- atingir flexibilidade: atingir modularidade: trabalho da Soeli?



**Figura 1: Modelo de referencia para ambiente acadêmico**

- Design rationale: modelo simplificado, foco naquilo que o estudante fez, na decisão tomada. Isso ajuda a entender o projeto na fase de manutenção. Tira foco daquilo que não fez (alternativas, posição).
- Estrutura da 15504: Process ID, Process name, process purpose, process outcome, base practices, work products. Estou incluindo na definição do processo academico: papéis, comentários, (sugestão da Rosly: critérios de entrada e métodos de saída).
- : Diferenciais do meu trabalho: processos para ambiente academico não foram pensando na forma de modelo de referencia ainda. Não foram considerados sob a perspectiva dos atributos de simplicidade e flexibilidade (acreditamos que considerar tais atributos é fundamental para motivar o uso na pratica). As propostas na literatura para desenvolvimento academico no geral visam o *ensino* e nao a *pesquisa*. Estas propostas estão voltadas para o desenvolvimento de software e não o desenvolvimento de projetos.
- Características do modelo de referencia que proponho: apresento uma tarefa, mas nao indico uma metodologia rígida que deve ser seguida para cumpri-la. Apresento indícios. indico referencias sempre que possível. Forneço comentários sugerindo bibliografia adicional.
- Artigo: enriquecendo o codigo com cenarios. Apresenta uma integraçao de documentacao e codigo fonte. Pg 161, anais sbes2003.
- Há uma tendência e um interesse dos pesquisadores em levar os resultados dos projetos academicos para o contexto industrial. (??) Como observado por Lucredio et al, em sua experiencia

com o desenvolvimento da MVCASE, isso exige o cumprimento de um processo de desenvolvimento de software (artigo da MVCASE). Outro artigo que mostra esta tendencia é aquele que fala na divulgação dos projetos. Lucredio tambem fala na divulgação.

- Ver projetos:
  - Ver outros projetos no repositorio Java, favoritos no explorer; tigris, relatorios tecnicos de documentacao de ferramentas do icmc.
  - Herbsleb (1999) descreve alguns problemas que prejudicam a comunicação em equipes distribuidas. Esta referencia está na pag. 125 dos anais do simpros 2004.
  - Procurar referencias do projeto prosoft

## Referências

[jun]

- [1] Amaral, D. C. (2001). *Arquitetura para Gerenciamento de Conhecimentos explícitos sobre Processo de Desenvolvimento de Produto*. PhD thesis, Universidade de São Paulo.
- [2] Ambati, V. and Kishore, S. P. (2004). How can Academic Software Research and Open Source Software Development Help Each Other? In *4th Workshop on Open Source Software Engineering, 26th International Conference on Software Engineering*, pages 5–8.
- [3] Bellotti, V., Burton, R., Ducheneaut, N., Howard, M., Neuwirth, C., and Smith, I. (2002). Xp in a research lab: The hunt for strategic value. In *Proceedings of the Third International Conference on eXtreme Programming (XP2002)*, pages 56–61.
- [Brito et al.] Brito, R., Ferreira, P., Silva, K., Burégio, V., and Leite, I. Uma Experiência na Implantação de Processo em uma Fábrica de Software Livre. In *VI Simpósio Internacional de Melhoria de Processos de Software*.
- [4] Chang, C. K. and Trubow, G. B. (1990). Joint Software Research Between Industry and Academia. *IEEE Software*, 7(6):71–77.
- [5] Chirouze, O., Cleary, D., and Mitchell, G. G. (2005). A Software Methodology for Applied Research: eXtreme Researching. *Software – Practice and Experience*, 35(15):1441–1454.
- [6] Esteves, R. P., Abalém, A. J. G., and Stanton, M. A. (2005). Algorithm for Establishing LSPs for Multicast Communication over GMPLS-based Networks. In *4th International Information and Telecommunication Technologies Symposium*.
- [7] et al, M. R. S. M. (2005). WQoS: Um Framework de alto nível para provisao de QoS em Sistemas Móveis sem Fio. In *4th International Information and Telecommunication Technologies Symposium*, pages 117–127.
- [Falbo] Falbo, R. A. Integração de Conhecimento em um Ambiente de Desenvolvimento de Software, school=Universidade Federal do Rio de Janeiro, year =1998,.
- [8] Goes, J. A. (2001). Perf: Ambiente de desenvolvimento e estimação temporal de sistemas de tempo real.

- [9] Goes, J. A. (2002). Adequas: Ferramenta fuzzy para avaliação da qualidade de software.
- [10] Goes, J. A. and Renaux, D. P. B. (2001). PERF: Um Ambiente de Desenvolvimento Voltado para Testes Temporais de Software. In *XV Simpósio Brasileiro de Engenharia de Software, XVIII Sessão de Ferramentas*, pages 404–409.
- [11] Guan, L. Z., Berzins, V., Zhang, L., Floodeen, D., V. C., Puett, J., and Brown, M. (2004). Requirements-document-based Prototyping of CARA Software. *International Journal on Software Tools for Technology Transfer*, 5(4):370–390.
- [12] Holmes, R. and Murphy, G. C. (2005). Using structural context to recommend source code examples. In *27th International Conference on Software Engineering*.
- [13] Ionescu, B., Ionescu, M., Veres, S., and Ionescu, D. (2005). Design and Architecture of an IP over Optical and Optical Ethernet Next Generation Network – NCIT\* net2. In *4th International Information and Telecommunication Technologies Symposium*.
- [14] Jenkins, M. (2004). PRO-SOFTWARE: A Government-Industry-Academia Partnership that Worked . In *17th Conference on Software Engineering Education and Training (CSEET'04)*, pages 92–97. IEEE Computer Society.
- [15] Junior, M. G. C., Sant'Anna, N., Filho, L. F. B., Genvigir, E. C., Luque, L., Tavares, R. P., and Casillo, B. H. (2003). (PDE) Process Definition Environment: Uma Ferramenta para Apoio à Definição de Processos de Software no Ambiente e-WebProject. In *XIV Congresso Internacional de Tecnologia de Software*, pages 42–55.
- [16] Kornecki, A., Khajenoori, S., Gluch, D., and Kameli, N. (2003). On a Partnership Between Software Industry and Academia. In *Proceedings of the 16th Conference on Software Engineering Education and Training*, pages 60–69.
- [17] Lavazza, L. and Barresi, G. (2005). Automated Support for Process-aware Definition and Execution of Measurement Plans. In *27th International Conference on Software Engineering*, pages 234–243.
- [18] Macke, S., New, J., Coxon, J., Manente, B., Khajenoori, S., Hirmanpour, I., and Ceberio, A. (1996). An Industry/Academic Partnership that Worked: an in Progress Report . In *Ninth Conference on Software Engineering Education*, pages 234–245. IEEE Computer Society.
- [19] Maidantchik, C. L. L. (1999). Gerência de Processos de Software para Equipes Geograficamente Dispersas.
- [20] Martin, J. C. (2004). Formal Methods Software Engineering for the CARA System. *International Journal on Software Tools for Technology Transfer*, 5(4):301–307.
- [21] Oliveira, H. L. R., Rocha, C. R. P., Gonçalves, K. M., and Souza, C. R. B. (2001). Utilização de Sistemas Críticos nas Atividades de Engenharia de Domínio e de Aplicações. In *XV Simpósio Brasileiro de Engenharia de Software*, pages 21–35.
- [22] Oliveira, K. R. and Belchior, A. D. (2003). Adequas: Ferramenta Fuzzy para Avaliação da Qualidade de Software. In *XVII Simpósio Brasileiro de Engenharia de Software, X Sessão de Ferramentas*, pages 31–36.

- [23] Reis, C. A. L. (2003). *Uma Abordagem Flexível para Execução de Processos de Software Evolutivos*. PhD thesis, Universidade Federal do Rio Grande do Sul.
- [24] Reis, R. Q. (1998). Uma Proposta de Suporte ao Desenvolvimento Cooperativo de Software no Ambiente PROSOFT.
- [25] Robillard, P. N. and Robillard, M. (1998). Improving Academic Software Engineering Projects: A Comparative Study of Academic and Industry Projects. *Annals of Software Engineering*, 6:343–363.
- [26] Rost, J. (2005). Software Engineering Theory in Practice. *IEEE Software*, 22(2):94–96.
- [27] Sante, D. G. (2003). AutorE: Suportando Autoria Evolucionária em Ambientes de Captura.
- [28] Silva, L. F., Sayão, M., Leite, J. C. S. P., and Breitman, K. K. (2003). Enriquecendo o Código com Cenários. In *XVII Simpósio Brasileiro de Engenharia de Software, Sessão de Ferramentas*, pages 161–176.
- [29] Sodre, R. A. (2001). Infra-estrutura de Segurança para Sistemas Distribuídos Baseada em Linguagem Formal.
- [30] Spindola, B., Albuquerque, C., Mascena, J., Coelho, K., Albuquerque, R., and Rocha, T. (2004). Definição e Melhoria de Processos em uma Fábrica de Software Livre. In *VI Simpósio Internacional de Melhoria de Processos de Software*, pages 189–200.
- [31] Succi, G. and Zuliani, P. (2004). Exploiting the Collaboration between Open Source Developers and Research. In *4th Workshop on Open Source Software Engineering, 26th International Conference on Software Engineering*, pages 97–99.
- [32] Zorzatto, J., Dodonov, E., Quaini, J., and Guardia, H. C. (2005). ProGrid Portal: Integrated Management and Security for Users, Resources and Applications. In *4th International Information and Telecommunication Technologies Symposium*.