

UNIVERSIDADE FEDERAL DE PERNAMBUCO  
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO  
CENTRO DE INFORMÁTICA

PROPOSTA DE TRABALHO DE GRADUAÇÃO

**Avaliando a metodologia PRO.NET em  
relação ao CMMI**

**Aluno:** Mauro La-Salette Costa Lima de Araújo (mscla@cin.ufpe.br)  
**Orientador:** André Luís de Medeiros Santos (alms@cin.ufpe.br)

Recife, 17 de Maio de 2005.

# 1 Contexto

A indústria de desenvolvimento de software tem experimentado, ao longo das últimas décadas, o que comumente convencionou-se denominar a *crise do software*. Quando da identificação desta crise em 1968, durante uma reunião do *NATO Science Comittee*, foi cunhado o termo “Engenharia de Software” para sintetizar a aspiração, naquele momento, de tornar a tarefa de produção de software uma disciplina da engenharia. Entretanto, como bem exposto por Brooks [2], existem dificuldades essencialmente atreladas à tarefa de produção de software e, décadas depois, a Engenharia de Software mostra-se incapaz de lidar, de maneira completa, com a natureza abstrata de seu principal produto. Dados de 2000 [6] mostram que: 23% dos projetos de software são cancelados antes de sua conclusão; 49% são concluídos ultrapassando os custos e prazos previstos; e apenas 28% são concluídos dentro dos custos e prazos previstos inicialmente.

Ao longo dos anos, a abordagem para dirimir as dificuldades presentes na tarefa de construção de artefatos de software, tem sido a de desenvolver ferramentas, tecnologias e processos que dêem apoio a esta tarefa. Enquanto ferramentas e tecnologias são responsáveis por aumentar a produtividade da equipe de desenvolvimento, o uso de processos visa especificar um conjunto de atividades que disciplinem todo o ciclo de desenvolvimento de software; numa tentativa de erradicar abordagens artesanais do mesmo. Alguns modelos de processo (*frameworks*), como o *Rational Unified Process* (RUP) [1], provêm um nível detalhado do fluxo das atividades a serem seguidas, enquanto outros, como o *Microsoft Solution Framework* (MSF) [4], fornecem apenas linhas mestras e recomendam boas práticas para a instanciação de um processo.

Mas o uso de um processo durante o desenvolvimento do software não é, por si só, suficiente para garantir o sucesso deste projeto. Muitas vezes, as organizações utilizam seus próprios processos de desenvolvimento que, quando não essencialmente ineficientes e artesanais, são executados de maneira *ad hoc*. E mesmo tentativas de utilizar processos reconhecidos e amplos como o RUP resultam em fracasso por dificuldades para adaptá-los às suas realidades. Destarte, vê-se que instanciar e/ou executar um processo de desenvolvimento de software é uma tarefa não-trivial. Assim, sob a influência destas dificuldades, tem-se procurado estabelecer um modelo de avaliação para o nível de maturidade dos processos de desenvolvimento de software das organizações.

Sob a iniciativa do *Software Engineering Institute* (SEI)<sup>1</sup>, a partir de uma requisição do governo federal dos EUA em meados da década de 1980, foi

---

<sup>1</sup><http://www.sei.cmu.edu/>

desenvolvido um modelo de avaliação do nível de maturidade do processo de desenvolvimento de software para organizações. Tal modelo foi denominado *Capability Maturity Model for Software* (SW-CMM ou simplesmente CMM) [9]. O CMM evoluiu durante os últimos anos e absorveu outros padrões de avaliação de processos como o *Electronic Industries Alliance Interim Standard* (EIA/IS), para engenharia de sistemas, e o *Integrated Product Development Capability Maturity Model* (IPD-CMM). Passando a chamar-se *Capability Maturity Model Integration* (CMMI) [3].

Durante a década de 1990, diversas organizações, impulsionadas pelo mercado extremamente competitivo, buscaram a adequação ao CMM como uma maneira de comprovar a qualidade do seu processo de desenvolvimento de software. E agora, com o advento do CMMI e descontinuidade do CMM, tais organizações estão realizando a migração entre os modelos. Assim, o CMMI evolui rapidamente para se tornar um padrão no que concerne à avaliação do nível do processo de desenvolvimento de software das organizações. A nova versão do MSF, por exemplo, apresentará práticas direcionadas ao CMMI.

## 2 Objetivos

A metodologia PRO.NET [5], desenvolvida no *Centro de Tecnologia XML* de Recife, foi concebida com base no RUP, no MSF, em conceitos de desenvolvimento ágil (*eXtreme Programming*) e é voltada para o desenvolvimento de projetos que utilizem a plataforma tecnológica .NET.

Um trabalho anterior realizou uma análise comparativa entre a metodologia PRO.NET e CMM-2 [8]. O presente trabalho terá por objetivo avaliar a metodologia PRO.NET em relação ao novo modelo CMMI, a fim de verificar as facilidades que a adoção da PRO.NET oferece para uma organização que pretende adequar-se ao CMMI. Utilizar-se-á o modelo de avaliação contínua do CMMI (por capacidades) para que a análise não se restrinja a um nível específico do modelo.

Também serão consideradas modificações na PRO.NET visando uma atualização em relação a nova versão do MSF. Mas isso dependerá, a priori, da disponibilidade da mesma, pois esta ainda não foi oficialmente lançada.

O produto do trabalho, além da análise comparativa, serão sugestões de melhoria na metodologia.

### 3 Cronograma

<i>Atividade</i>	<i>Mês</i>											
	<i>Mai</i>			<i>Junho</i>			<i>Julho</i>			<i>Agosto</i>		
Estudar a metodologia PRO.NET e o MSF	*	*	*									
Estudar os modelos CMM e CMMI				*	*	*						
Elaborar avaliação comparativa							*	*	*	*	*	*
Elaborar o relatório			*	*	*	*	*	*	*	*	*	*
Elaborar a apresentação											*	*

Tabela 1: Cronograma

### Referências

- [1] G. Booch, I. Jacobson, and J. Rumbaugh. *The Unified Software Development Process*. Addison-wesley, 1999.
- [2] F. P. Brooks. *No silver bullet: Essence and accidents of software engineering*. *Computer Magazine*, April 1987.
- [3] Mary Beth Chrissis, Mike Konrad, and Sandy Shrum. *CMMI: Guidelines for Process Integration and Product Improvement*. Addison Wesley, February 2003.
- [4] Microsoft Corp. *Microsoft solutions framework version 3.0 overview*, July 2003.
- [5] CTXML. Site contendo toda a metodologia PRO.NET. Disponível em <http://www.cin.ufpe.br/~pronet>.
- [6] Standish Group. 2001 update to the chaos report, 2001. Disponível em: [http://www.standishgroup.com/sample\\_research/](http://www.standishgroup.com/sample_research/).
- [7] IEEE. *IEEE standard 610.12-1990 IEEE standard glossary of software engineering terminology*, 1990.
- [8] Suzana Maranhão. *Análise comparativa entre a pro.net e o cmm nível 2*, August 2003.
- [9] M. C. Paulk, B. Curtis, M. B. C. Chrissis, and C. Weber. *Capability maturity model for software, version 1.1*. Technical Report ADA263403, Software Engineering Institute, February 1993.

## Datas e Assinaturas

17 de Maio de 2005

---

André Luís de Medeiros Santos  
(Orientador)

---

Mauro La-Salette Costa Lima de Araújo  
(Proponente)