

Inspeções de Requisitos de Software em Desenvolvimento Incremental: Uma Experiência Prática

Marcos Kalinowski¹, Rodrigo Oliveira Spínola¹, Arilo Claudio Dias Neto¹,
Andrew Bott², Guilherme Horta Travassos¹

¹Programa de Engenharia de Sistemas e Computação – COPPE/UFRJ
Caixa Postal 15.064 – 91.501-970 – Rio de Janeiro – RJ – Brazil.

²Fundação Coordenação de Projetos, Pesquisas e Estudos Tecnológicos – COPPETEC
Ilha do Fundão – 21949-900 – Rio de Janeiro – RJ – Brazil.

{mkali, ros, acdn, ght}@cos.ufrj.br, andrew@coppetec.coppe.ufrj.br

Abstract. *Software inspection is an efficient technique to reduce rework and improve the quality of software artifacts. However, in industry inspections are often conducted in a non-systematic way and their full potential is seldom achieved. In this paper we report the experience of institutionalizing the software inspection process in order to accomplish requirements inspections in a real project with an incremental life cycle. Among the obtained results we highlight the identification of opportunities for improving the software process, based on the analysis of the inspection results previous increments.*

Resumo. *Inspeção de software é uma técnica eficiente para reduzir o retrabalho e melhorar a qualidade de artefatos de software. Entretanto, na indústria a aplicação de inspeções muitas vezes se dá de forma pouco sistemática e seu verdadeiro potencial é raramente atingido. Neste artigo relatamos a experiência da institucionalização do processo de inspeção de software para a realização de inspeções de requisitos em um projeto real que segue um ciclo de vida incremental. Entre os resultados obtidos destacamos a identificação de oportunidades de melhoria no processo de software, a partir da análise dos resultados das inspeções em incrementos anteriores.*

1. Introdução

Sabe-se que o custo da correção de defeitos aumenta na medida que o processo de desenvolvimento progride (Boehm, 1981) (Boehm e Basili, 2001). Revisões de artefatos de software têm se mostrado uma abordagem eficiente e de baixo custo para encontrar defeitos logo após terem sido introduzidos, reduzindo o retrabalho e melhorando a qualidade dos produtos (Ciolkowski *et al.*, 2002). Não é em vão que modelos de maturidade de processo de software, como o CMMI (Crissis *et al.*, 2006) e o MPS (Softex, 2007), exigem a condução de revisões.

Um tipo particular de revisão de software são as inspeções (Fagan, 1976). Inspeções possuem um processo de detecção de defeitos rigoroso e bem definido. A importância de inspeções na garantia da qualidade de software está bem documentada na literatura (Boehm e Basili, 2001) (Ciolkowski *et al.*, 2002). Os benefícios de se aplicar inspeções são maiores para artefatos desenvolvidos no início do processo, como o documento de requisitos (Shull *et al.*, 2001).

O processo de inspeção tradicional (Fagan, 1976) envolve o planejamento da inspeção, a revisão de um determinado artefato por indivíduos previamente selecionados, um encontro em equipe para discutir e registrar os defeitos, a passagem dos defeitos para o autor do artefato para que possam ser corrigidos e uma avaliação final sobre a necessidade de uma nova inspeção.

Ao longo dos anos muitas teorias e técnicas relacionadas com inspeções de software têm sido propostas. Algumas das quais foram avaliadas experimentalmente e

são consideradas parte do corpo de conhecimento na área de inspeções de software. Este conhecimento envolve reorganizações do processo (Sauer *et al.*, 2000), heurísticas para a escolha de inspetores de acordo com suas características (Carver, 2003), técnicas de estimativa do número de defeitos em documentos (Biffi *et al.*, 2003) e da eficiência das inspeções como um todo (Adams, 1999), técnicas de leitura e *checklists* (Travassos *et al.*, 1999) (Shull *et al.*, 2001) (Mafrá e Travassos, 2006) (Barcelos e Travassos, 2006), entre outros.

Entretanto, resultados de um *survey* (Ciolkowski *et al.*, 2003) mostram que embora muitas empresas realizem revisões de software, elas o fazem de forma pouco sistemática e pouco conhecimento a respeito de inspeções de software é utilizado. Assim, o verdadeiro potencial das inspeções é raramente atingido. Por isto, apoio ferramental surgiu para viabilizar a institucionalização do processo de inspeção de software de forma sistemática, automatizando, entre outros, a coordenação do processo de inspeção.

Neste artigo relatamos a experiência da institucionalização do processo de inspeção aplicada a requisitos de software, utilizando apoio ferramental, em um projeto de software em parceria com a indústria, seguindo um ciclo de vida incremental.

O restante do artigo está organizado da seguinte forma. Na seção 2 o projeto e o processo utilizado no desenvolvimento são descritos, destacando detalhes do processo de inspeção. Na seção 3, a experiência de conduzir inspeções de requisitos é relatada. Na seção 4 uma análise da experiência é apresentada, destacando dados quantitativos e qualitativos da mesma. Por fim, na seção 5, as considerações finais são descritas.

2. Caracterização do Projeto e do Processo de Inspeção

O projeto referente a este relato de experiência visa o desenvolvimento de um novo sistema de informação para gerenciamento das atividades da Fundação COPPETEC. Trata-se de um projeto de grande porte que envolve diferentes setores da organização, como: recursos humanos, financeiro, contabilidade e protocolo. O projeto pôde ser modularizado e desenvolvido seguindo um ciclo de vida incremental. Entre os fatores que justificaram esta decisão, podemos citar os dois seguintes: o cliente está interessado em (1) entregas parciais do produto e (2) na substituição gradual do sistema de informação atual pelo novo.

A partir do ciclo de vida a ser utilizado (incremental), definiu-se o processo de desenvolvimento do software, fortemente fundamentado em atividades de Verificação, Validação e Testes (VV&T). As atividades de VV&T incluídas no processo são: inspeções, testes estruturais, testes funcionais e testes beta (de aceitação). Um gerente de VV&T foi definido para viabilizar o planejamento e acompanhamento destas atividades.

Considerando as inspeções, o apoio ferramental ISPIS (Kalinowski *et al.*, 2004) foi utilizado para coordenar sua condução. ISPIS embute conhecimento experimentalmente avaliado e tem sido explorada experimentalmente na academia (Kalinowski e Travassos, 2004) e na indústria (Kalinowski e Travassos, 2005). O processo utilizado nas inspeções está ilustrado na Figura 1 (Kalinowski *et al.*, 2004). Segue uma descrição resumida das atividades do processo.

- **Planejamento.** Um usuário desempenhando o papel de moderador da inspeção define o contexto da inspeção (descrição da inspeção, técnica a ser utilizada na detecção de defeitos, documento a ser inspecionado, autor do documento, entre outros), seleciona os inspetores e distribui o material a ser inspecionado.
- **Detecção de Defeitos.** Cada um dos inspetores selecionados pelo moderador realiza a detecção de defeitos. A principal tarefa desta atividade consiste em buscar defeitos no documento a ser inspecionado, produzindo uma lista de discrepâncias.

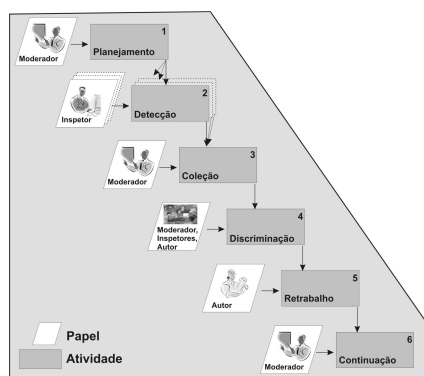


Figura 1. Processo de Inspeção.

- **Coleção de Defeitos.** O moderador agrupa as listas de discrepâncias dos inspetores e elimina discrepâncias repetidas (encontradas por mais de um inspetor), mantendo só um registro para cada discrepância.
- **Discriminação de Defeitos.** O moderador, o autor do documento e os inspetores discutem as discrepâncias. Durante esta discussão, as discrepâncias serão classificadas como falso positivo ou defeito. Os falsos positivos serão descartados e os defeitos serão registrados em uma lista de defeitos.
- **Retrabalho.** O autor do documento corrige os defeitos da lista de defeitos e produz um relatório de correção dos defeitos.
- **Continuação.** O moderador decide se uma nova inspeção deve ou não ocorrer.

Os dados obtidos ao longo de cada inspeção são mantidos em uma base histórica do projeto. No cenário do processo incremental descrito no início desta seção, manter esses dados, como ocorre naturalmente em ISPIS, se mostra particularmente interessante, pois eles podem servir como ponto de partida para a melhoria de processo através de análise causal e resolução de defeitos de software, atividade que se encontra nos níveis de maturidade A do MPS e 5 do CMMI.

A seção seguinte descreve a experiência da instanciação do processo de inspeção acima descrito para inspeções nos requisitos dos diferentes módulos do novo sistema de informação.

3. Aplicando Inspeções de Requisitos

Inspeções de requisitos foram conduzidas para cada um dos módulos do sistema a ser desenvolvido. No momento da escrita deste artigo, três inspeções encontram-se concluídas e uma em andamento. As inspeções concluídas foram, em ordem cronológica, referentes aos módulos MSL (módulo de solicitações), MGU (módulo de gestão de usuários) e MPT (módulo de protocolo). A inspeção em andamento, referente ao módulo MPTV (módulo de protocolo para projetos vinculados) acaba de ser planejada e ainda não possui resultados relatáveis.

Para institucionalizar o processo de inspeção, o apoio ferramental ISPIS foi instalado e treinamento separado foi fornecido para participantes dos diferentes papéis: moderadores, inspetores e autores. Cada inspeção envolveu um conjunto diferente de pessoas e ao definir a equipe de inspeção (autor do documento e inspetores) para um determinado módulo, o gerente de VV&T (que exerceu o papel de moderador) verificava se todos os participantes já haviam recebido treinamento. No treinamento, com duração de 1 hora, uma visão geral do processo e a forma de realizar as atividades referentes ao papel que estava desempenhando eram apresentadas. A seguir, encontra-se uma descrição de como a execução das atividades do processo se deu em nossa experiência.

Planejamento. Nessa atividade, o moderador visava, de forma estratégica, envolver inspetores com perspectivas complementares na identificação de defeitos. Assim, as inspeções faziam uso de inspetores representando os diferentes consumidores do documento de requisito, e em cada uma delas havia pelo menos um membro da equipe de desenvolvimento, um profissional responsável pelo planejamento dos testes e um representante do cliente.

Detecção de Defeitos. Durante essa atividade, cada um dos inspetores tinha acesso ao

documento a ser inspecionado e relatava as discrepâncias encontradas em ISPIS preenchendo as seguintes informações: localização no documento, classificação, severidade e descrição. A classificação seguia a taxonomia proposta por Shull (1998), envolvendo os seguintes tipos de defeito: omissão, ambigüidade, inconsistência, fato incorreto, informação estranha e outro tipo de defeito. A severidade, por sua vez, assumia os seguintes valores: baixa, média, alta e não definida.

Coleção de Defeitos. O moderador então, na atividade de coleção de defeitos, identificava as discrepâncias encontradas por mais de um inspetor e organizava a lista de discrepâncias a serem discutidas pela equipe de inspeção na atividade de discriminação de defeitos. Utilizando o conhecimento descrito em (Lanubile e Mallardo, 2003), as discrepâncias encontradas por mais de um inspetor foram encaminhadas diretamente ao retrabalho pelo autor.

Discriminação de Defeitos. Nessa atividade, a equipe discutia as discrepâncias encontradas de forma assíncrona. Comentários eram acrescentados a respeito das mesmas de forma anônima e o moderador podia classificá-las como defeito ou descartá-las como falso-positivos. A literatura sugere inspeções com reuniões assíncronas para reduzir custos e conflitos de alocação de recursos para reuniões. Entretanto, na nossa experiência notamos que disponibilizar as discrepâncias para discussão assíncrona resultou em pouca participação dos representantes do cliente envolvidos na inspeção. Este dado, baseado no número de comentários, nos sugere experimentar o uso de reuniões síncronas para os próximos módulos.

Retrabalho. Na seqüência, o autor do documento recebia a lista de defeitos para correção e registrava as correções feitas na atividade de retrabalho.

Continuação. Após a conclusão do retrabalho, o moderador avaliava a qualidade da inspeção na atividade de continuação. Para esta avaliação ISPIS provê acesso ao formulário de correção de defeitos e as estimativas da eficiência da inspeção (Biffi *et al.*, 2003) e de uma possível re-inspeção (Adams, 1999). Para os módulos MSL, MGU e MPT uma inspeção foi considerada suficiente.

A Figura 2 ilustra parte do formulário de correção de defeitos para a inspeção do módulo MSL, inspeção em que 60 defeitos foram encontrados em um documento de requisitos de 43 páginas. No módulo MGU foram encontrados 16 defeitos em um documento de requisitos de 19 páginas. Por fim, no MPT foram encontrados 63 defeitos em um documento de requisitos de 108 páginas.

Defeito 16	
Localização	RS - Página 11
Classificação	Informação Inconsistente
Severidade	Baixa
Descrição	Na descrição dos casos de uso, para um mesmo "tipo" de condição, ora há pós-condição ora não há, por exemplo, cadastramento com sucesso de um item.
Comentário de Correção	CORRIGIDO Do UC07 ao UC17 não tinha pós-condição.
Comentários (3)	
1. Moderador	Não entendi direito qual o problema, poderia especificar mais?
2. Inspetor 2	Sim, o que eu quis dizer é que uma pós condição do tipo "o sistema armazena as modificações com sucesso" em alguns casos aparece e outras, não. Se "cadastrar informação com sucesso" não é uma pós-condição "relevante" acredito que nenhum caso de uso deveria ter tal "tipo" de pós-condição.
3. Moderador	OK, sugiro que tentemos padronizar isto, vou classificar como defeito bom baixa severidade.

Figura 2. Parte do relatório de correção de defeitos do módulo MSL extraído de ISPIS.

Considerando os defeitos encontrados, pudemos observar que alguns poderiam resultar em sérios problemas para o cliente, caso fossem implementados da forma como estavam originalmente descritos. Dois exemplos de tais defeitos, encontrados no módulo MPT, se encontram descritos a seguir. A identificação destes defeitos só foi possível através do uso de inspetores especialistas no domínio da organização cliente.

Defeito 1. Durante o levantamento de requisitos foi identificado que para protocolar uma determinada solicitação, o sistema deveria gerar um número de protocolo. Durante a inspeção identificou-se que além do número de protocolo gerado para a solicitação como um todo, era necessário que o sistema gerasse um conjunto seqüencial de números de protocolo considerando as notas fiscais anexadas à solicitação.

Defeito 2. Durante o levantamento de requisitos foi identificado que existia um cadastro único de plano de contas de onde suas informações poderiam ser recuperadas. Durante a inspeção, identificou-se que existiam na verdade duas definições de plano de contas: uma, *mais detalhada*, considerando a classificação utilizada internamente pela Fundação COPPETEC, e outra, mais sintética, para atender o padrão de classificação exigido no âmbito de projetos desenvolvidos para órgãos do governo.

Relatada a experiência, na seção seguinte se encontra uma análise baseada em dados quantitativos obtidos a partir das inspeções.

4. Análise da Experiência

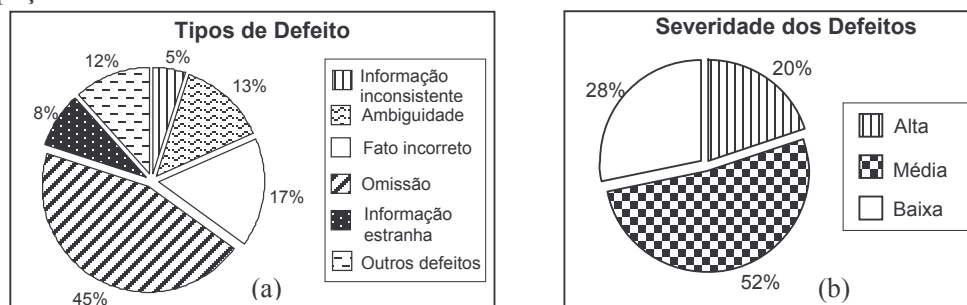
Encontram-se a seguir uma análise a respeito das inspeções de requisitos nos módulos MSL, MGU e MPT, e por fim de uma forma generalizada.

4.1. Resultados das Inspeções de Requisitos no Módulo MSL

Neste módulo, a equipe de inspeção foi formada por 1 moderador, 1 autor e 5 inspetores (2 especialistas do domínio do lado cliente, 2 projetistas/desenvolvedores e 1 planejador de testes). O planejamento da inspeção teve duração aproximada de 1 hora. O prazo para a detecção de defeitos por parte dos inspetores foi de 4 dias úteis. Este prazo considera a agenda e disponibilidade de toda a equipe de inspeção. Considerando os inspetores da equipe do projeto, durante este prazo eles foram envolvidos em atividades sendo desenvolvidas em paralelo para outros incrementos. Por sua vez, os inspetores representantes do cliente estavam envolvidos em outras atividades da instituição.

O esforço médio efetivamente gasto pelos inspetores foi de aproximadamente 6 horas. A atividade de coleção levou 2 horas. A atividade de discriminação ficou disponível por 3 dias úteis, e envolvia pouco esforço, uma vez que a equipe era notificada por e-mail sempre que um comentário era adicionado a uma discrepância, podendo responder aos comentários para colaborar com a correta classificação desta pelo moderador. Entretanto, mesmo desempenhando outras atividades durante a inspeção, o andamento do projeto arquitetural correspondente aos requisitos sendo inspecionados poderia ser prejudicado caso aguardássemos o término da inspeção. Para lidar com este problema, assumimos o risco de retrabalho. Isto se mostrou vantajoso uma vez que o retrabalho afetou pouco o projeto arquitetural.

Comparando a eficiência dos inspetores, foi possível notar a total ausência de um dos especialistas do domínio. Dos quatro inspetores restantes, todos tiveram contribuições importantes, embora um tenha relatado aproximadamente três vezes mais defeitos do que os outros (38 defeitos dele contra 11, 12 e 14 dos demais). No total, 60 defeitos distintos foram encontrados no documento de 43 páginas. As Figuras 3a e 3b representam a distribuição dos tipos e severidade de defeitos, considerando a equipe de inspeção como um todo.

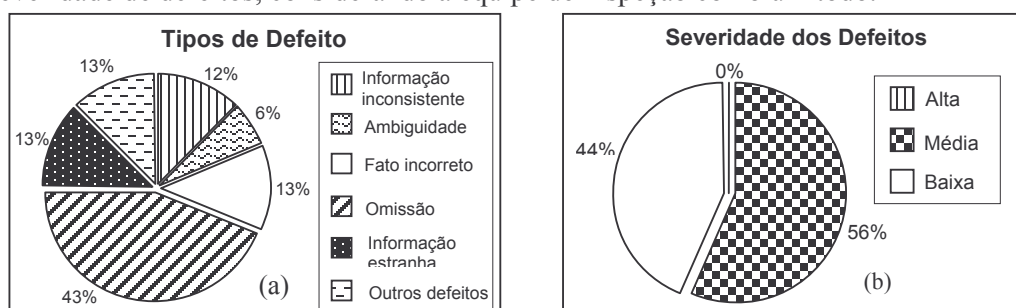


Figuras 3a e 3b. Distribuição dos tipos de defeito e sua severidade no módulo MSL.

4.2. Resultados das Inspeções de Requisitos no Módulo MGU

Neste módulo, a equipe de inspeção foi formada por 1 moderador, 1 autor e 3 inspetores (1 especialista do domínio do lado cliente, 1 projetista/desenvolvedor e 1 planejador de testes). O planejamento da inspeção novamente teve duração aproximada de 1 hora. O prazo para a detecção de defeitos por parte dos inspetores foi de 4 dias úteis e o esforço médio por inspetor foi de aproximadamente 3 horas. A atividade de coleção levou 1 hora e discriminação ocorreu dentro do mesmo dia útil.

A contribuição dos inspetores na inspeção do MGU foi mais bem distribuída (encontrando 5, 9 e 11 defeitos). No total 16 defeitos distintos foram encontrados no documento de 19 páginas. As Figuras 4a e 4b representam a distribuição dos tipos e severidade de defeitos, considerando a equipe de inspeção como um todo.

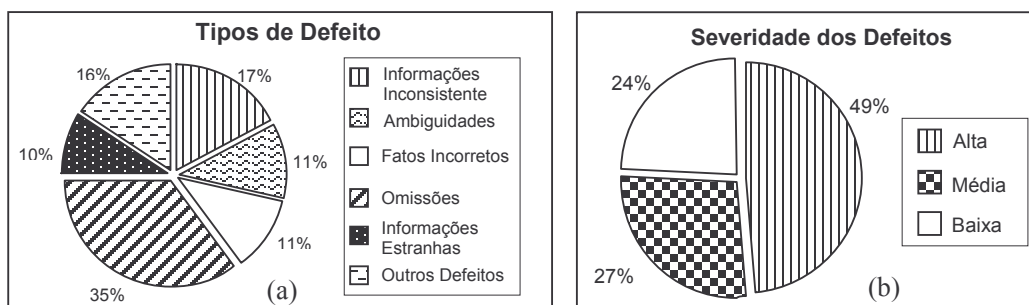


Figuras 4a e 4b. Distribuição dos tipos de defeito e sua severidade no módulo MGU.

4.3. Resultados das Inspeções de Requisitos no Módulo MPT

Neste módulo, a equipe de inspeção foi formada por 1 moderador, 1 autor e 4 inspetores (2 especialistas no domínio pelo lado cliente, 1 projetista/desenvolvedor e 1 planejador de testes). O planejamento da inspeção teve duração aproximada de 1 hora. O prazo planejado para a detecção de defeitos por parte dos inspetores foi de 5 dias úteis, mas este prazo acabou sendo estendido para 10 dias úteis por falta de disponibilidade imediata de um dos especialistas do domínio do cliente. O esforço médio gasto por inspetor na detecção de defeitos foi de aproximadamente 5 horas. A coleção ocorreu em 4 horas e a discriminação ficou disponível por 5 dias úteis.

Novamente todos os inspetores deram contribuições importantes (encontrando 8, 9, 24 e 27 defeitos). No total 63 defeitos distintos foram encontrados no documento de 108 páginas. As Figuras 5a e 5b representam a distribuição dos tipos de defeito e de sua severidade, considerando a equipe de inspeção como um todo.



Figuras 5a e 5b. Distribuição dos tipos de defeito e sua severidade no módulo MPT.

4.4. Análise Geral dos resultados das Inspeções de Requisitos no Projeto

Foi possível observar uma tendência do projeto em relação à distribuição dos tipos de defeitos. Após as inspeções dos módulos MSL e MGU, notamos, a partir do número alto de omissões, uma deficiência no processo de elicitação de requisitos. A mudança

foi passar a entrevistar além de pessoas da instituição cliente com perfil operacional, também pessoas com perfil gerencial. Assim, a proporção de defeitos do tipo omissão foi menor para os requisitos do módulo MPT. Entretanto, a distribuição entre os diferentes tipos de defeitos encontrados nas três inspeções ainda se mostra bastante parecida, como pode ser observado comparando as figuras 3a, 4a e 5a.

Outra melhoria observada foi a redução da densidade de defeitos (defeitos por página) entre as diferentes inspeções. A densidade passou de 1,4 no MSL para 0,84 no MGU e para 0,58 no MPT. Além da mudança no processo de elicitação de requisitos, acreditamos que esta redução tenha resultado do aprendizado adquirido pela equipe a respeito de defeitos de requisitos ao participar das inspeções, bem como o aprendizado do cliente em fornecer as informações necessárias para a especificação. O aprendizado ao longo de inspeções é também mencionado em (Shull, 1998).

Em relação à distribuição da severidade de defeitos, por sua vez, notamos um aumento do número de defeitos de severidade alta no módulo MPT. Isto se deve a percepções diferentes dos autores em relação ao domínio do problema, gerando incompatibilidades entre as especificações do MPT e dos demais módulos, que se foram classificados como defeitos de severidade alta. É importante ressaltar que o critério para a atribuição de severidade a defeitos foi subjetivo e que estes defeitos poderiam ter sido classificados como defeitos de severidade média.

5. Considerações Finais

Relatamos a experiência da institucionalização de inspeções de requisitos de software, utilizando apoio ferramental, em um projeto real de desenvolvimento de software seguindo um ciclo de vida incremental.

De maneira geral, as inspeções nos permitiram encontrar um número grande de defeitos no início do desenvolvimento de cada um dos módulos, momento em que a correção destes defeitos apresenta menor custo. Foram encontrados defeitos de severidade alta, que poderiam ter um grande impacto no projeto. Muitos destes defeitos foram encontrados por especialistas do domínio. Vale ainda ressaltar que nestas inspeções não foram utilizadas técnicas de leitura específicas nem *checklists*, recursos que têm mostrado aumentar a eficiência de inspetores. Consideramos coerente primeiro institucionalizar o processo de inspeção, ficando o aprimoramento da técnica de detecção de defeitos como uma oportunidade de melhoria futura.

O custo da realização das inspeções se mostrou baixo. Mesmo nas atividades de maior duração, como a detecção e a discriminação de defeitos, embora tais atividades ficassem disponíveis por períodos longos, o esforço dedicado a elas era baixo. O tempo médio dedicado por um inspetor à detecção de defeitos foi de menos de 5 horas. Considerando a discussão assíncrona da atividade de discriminação, notamos que o uso de um processo assíncrono baseado em apoio ferramental se mostra difícil quando o representante do cliente se encontra pouco disponível e participa pouco da atividade. Uma alternativa seria realizar reuniões síncronas, embora isto possa acarretar conflitos de alocação de recursos.

Acreditamos ainda que o apoio ferramental tenha facilitado a institucionalização do processo, uma vez que a ferramenta ISPIS é responsável pela a coordenação do processo e oferece apoio para a realização das diferentes atividades. O uso deste apoio permitiu a coleta automática de todos os dados da inspeção. A análise de tais dados, no contexto do processo de desenvolvimento seguindo um ciclo de vida incremental, nos permitiu observar tendências e identificar oportunidades de melhoria que serviram como ponto de partida para a melhoria do processo de elicitação de requisitos.

Agradecimentos

Os autores gostariam de agradecer a todos os profissionais que participaram das inspeções descritas neste artigo. Agradecemos ainda à CAPES, CNPq e FAPEAM pelo apoio financeiro.

Referências

- Adams, T. (1999), “A formula for the re-inspection decision”, *Software Engineering Notes* 24(3): 80.
- Barcelos, R.F., Travassos, G.H. (2006), “ArqCheck: Uma abordagem para inspeção de documentos arquiteturais baseada em checklist”, In: V Simpósio Brasileiro de Qualidade de Software, Vila Velha-ES.
- Biffi, S., Halling, M., Koeszegi, S. (2003), “Investigating the Accuracy of Defect Estimation Models for Individuals and Teams Based on Inspection Data”, *Proceedings of the 2nd International Symposium on Empirical Software Engineering*, Rome, Italy.
- Boehm, B. W., Basili, V.R. (2001), “Software Defect Reduction Top 10 List.”, *IEEE Computer* 34 (1): 135-137.
- Boehm, B. (1981), *Software Engineering Economics*, Prentice Hall.
- Carver, J. (2003), “The Impact of Background and Experience on Software Inspections”, PhD Thesis, University of Maryland, USA.
- Ciolkowski, M., Laitenberger, O., Rombach, D., Shull, F., Perry, D. (2002), “Software inspections, reviews & walkthroughs”, *Proceedings of the 24th International Conference on Software Engineering*, Orlando, Florida.
- Ciolkowski, M., Laitenberger, O., Biffi, S. (2003), “Software Reviews: The State of the Practice”, *IEEE Software* 20 (6): 46-51.
- Crissis, M.B., Konrad, M., Shrum, S. (2006), *CMMI – Guidelines for Process Integration and Product Improvement (2nd Edition)*, SEI Series in Software Engineering, Addison-Wesley Professional.
- Fagan, M.E. (1976), “Design and Code Inspection to Reduce Errors in Program Development”, *IBM Systems Journal*, vol. 15, no. 3, pp. 182-211.
- Kalinowski, M., Spinola, R.O., Travassos, G. H. (2004) “Infra-Estrutura Computacional para Apoio ao Processo de Inspeção de Software”, In: III Brazilian Symposium on Software Quality, Brasília – DF, Brazil.
- Kalinowski, M., Travassos, G. H. (2004) “A Computational Framework for Supporting Software Inspections”, In: 19th IEEE International Conference on Automated Software Engineering - ASE'04, pp. 46-55, Linz, Austria.
- Kalinowski, M., Travassos, G.H. (2005) “Software Technologies: The Use of Experimentation to Introduce ISPIS - a Software Inspection Framework - Into the Industry”, In: *Experimental Software Engineering Latin American Workshop (ESELAW 2005)*, Uberlândia-MG, Brazil.
- Lanubile, F. Mallardo, T. (2003), “An Empirical Study of Web-Based Inspection Meetings”, *Proc. of the 2nd International Symposium on Empirical Software Engineering*, Rome, Italy.
- Mafra, S.N., Travassos, G.H. (2006), “Leitura Baseada em Perspectiva: A Visão do Projetista Orientada a Objetos”, In: V Simpósio Brasileiro de Qualidade de Software, Vila Velha-ES.
- Sauer, C., Jeffery, D.R., Land, L., Yetton, P. (2000), “The Effectiveness of Software Development Technical Review: A Behaviorally Motivated Program of Research”, *IEEE Transactions on Software Engineering*, 26 (1): 1-14, January.
- Shull, F., Rus, I., Basili, V. (2000), “How Perspective-Based Reading Can Improve Requirements Inspections”, July, *IEEE Software*, pp. 73-79.
- Shull, F. (1998), “Developing Techniques for Using Software Documents: A Series of Empirical Studies”, Ph.D. thesis, University of Maryland, College Park.
- Softex (2007), “Guia Geral do MPS.BR – Melhoria de Processo do Software Brasileiro”, disponível em <http://www.softex.br/mpsbr>.
- Travassos, G. H., Shull, F., Fredericks, M., Basili, V. R. (1999), Detecting Defects in Object Oriented Designs: Using Reading Techniques to increase Software Quality. *ACM Sigplan Notices*. Estados Unidos, v.34, n.10, p.47 - 56.