



Capítulo 6: PSP

Personal Software Process

- Capítulo 1: Introdução
- Capítulo 2: Conceitos Básicos
- Capítulo 3: Qualidade de Produto (ISO9126)
- Capítulo 4: ISO9001 e ISO9000-3
- Capítulo 5: CMM
- **Capítulo 6: PSP**
- Capítulo 7: SPICE
- Capítulo 8: Conclusão



Como chegamos até aqui

- Qualidade de produto de software (ISO 9126)
- Sistemas da Qualidade (ISO 9001 e 9000-3)
- Capabilidade e maturidade dos processos de software (CMM)
- Críticas: difícil aplicação em pequenas equipes de desenvolvimento ou no nível individual
 - CMM afeta em muito pouco as práticas pessoais de engenharia de software dos desenvolvedores
- Proposta de Watts Humphrey: PSP, o Personal Software Process



Objetivos do PSP

- Ajudar as pessoas a serem melhores engenheiros de software
- Estabelecer um mecanismo para melhorar, no nível pessoal, a capacidade de planejamento, acompanhamento e qualidade dos resultados
- Conceitos básicos do PSP podem ser usados como ferramenta de uso geral para gerenciar as atividades pessoais particulares ou profissionais
- Benefícios:
 - melhoria da produtividade: melhor conhecimento e controle dos mecanismos e tempos de produção
 - qualidade dos produtos: resultado do conhecimento das causas dos erros e do seu controle estatístico



Proposta do PSP

- Integrar-se às práticas organizacionais do CMM
 - permitir que a base da estrutura da organização, os desenvolvedores, tenham melhor desempenho pessoal
- Processos no nível pessoal também têm que ser conhecidos, controlados e melhorados
- PSP pode ser usado por pessoas em empresas que ainda estão no nível 1
 - alguns benefícios mútuos não aparecerão mas certamente a pessoa perceberá as melhorias no nível pessoal



Compromissos

- Como fazer para que compromissos assumidos pela alta gerência sejam transformados em compromissos pessoais dos desenvolvedores?
 - as pessoas só assumem compromissos pessoais voluntariamente
 - imposições não são compromissos; podem ser até aceitas, porém como obrigações e não como compromissos
- Cronogramas e planos corporativos podem não ser vistos pelos desenvolvedores como compromisso pessoal



PSP e os compromissos

- O PSP pode auxiliar neste processo desde que a visão dos desenvolvedores de como será a sua contribuição no desenvolvimento como um todo seja levada em conta
 - participação no processo de planejamento
 - as pessoas passam a conhecer melhor as suas capacidades e desempenho, o planejamento pessoal pode ser gradativamente integrado ao planejamento da empresa



A abordagem do PSP

- Identificar os métodos e práticas usados em grandes sistemas que podem ser usados por indivíduos
- Definir um subconjunto destes métodos e práticas que podem ser usados no desenvolvimento de pequenos programas
- Estruturar e escalonar estes métodos e práticas de modo a possibilitar a sua introdução gradual e disciplinada
- Desenvolver exercícios para facilitar a introdução das práticas

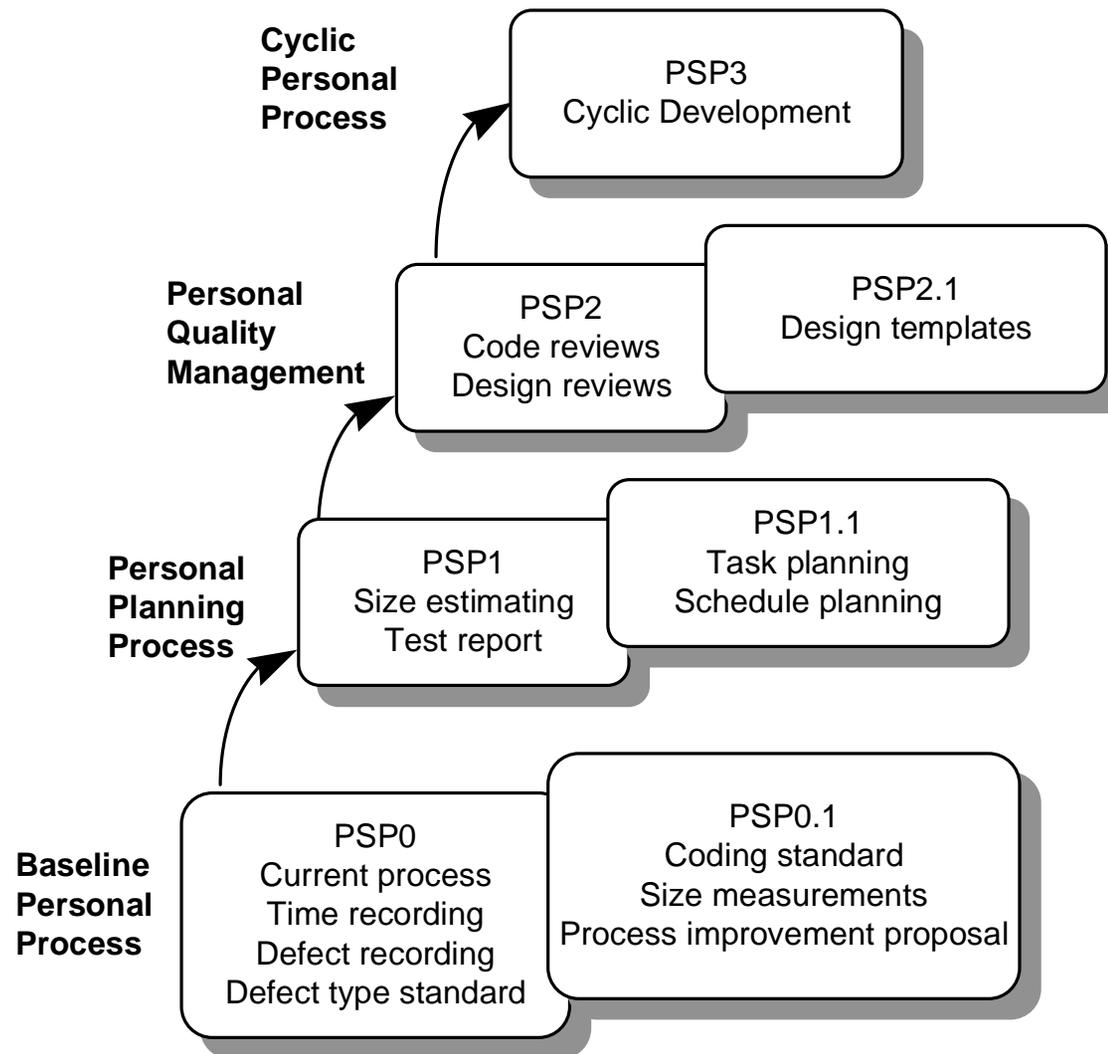


KPAs não implementadas pelo PSP

- SSM e Coordenação Entre Grupos (Intergroup Coordination):
 - não podem ser praticadas no nível individual
- RM e SCM :
 - poderiam ser aplicadas no nível individual mas devem ser consideradas apenas após a implantação dos passos iniciais do PSP
- SQA e Programa de Treinamento:
 - referem-se mais aos aspectos organizacionais
 - como a organização verifica o atendimento aos procedimentos estabelecidos
 - como a organização planeja e executa a evolução da capacitação do seu pessoal



Os processos do PSP





PSP0: Baseline Process

- PSP0: estabelecimento de práticas de medida e alguns formatos de relatórios que constituirão uma base (baseline) ou fundação sobre a qual será implantada a melhoria contínua pessoal
 - não afeta métodos e práticas pessoais de design, codificação e teste (apenas serão medidos)
- O PSP0.1 acrescenta padrões de programação, práticas de medida de tamanho de produto de trabalho e o Process Improvement Proposal (PIP — Proposta de Melhoria de Processo)
 - PIP é uma forma estruturada de registrar problemas nos processos, experiências e propostas de melhoria



PSP1: Processo de Planejamento Pessoal

- O PSP1 acrescenta práticas de planejamento ao PSP0
- Inicialmente, apenas relatórios de teste e práticas de estimativa de tamanho e recursos
- Em seguida, no PSP1.1, planejamento de tarefas e elaboração de cronogramas são introduzidos
- O planejamento no nível pessoal possibilita:
 - melhor compreensão da relação entre o tamanho dos programas e o tempo gasto no seu desenvolvimento
 - assumir compromissos com mais certeza de que serão cumpridos
 - organizar o trabalho
 - melhor acompanhamento do status do desenvolvimento



PSP1: Processo de Planejamento Pessoal (cont.)

- Objetivos de planejamento são importantes não só para projetos de grandes dimensões, mas também para indivíduos que trabalham sozinhos no desenvolvimento de sistemas
- Se a produtividade do indivíduo é bem conhecida, ele pode planejar melhor o seu trabalho, assumir compromissos com mais segurança e cumprí-los com maior regularidade



PSP2: Processo de Gestão Pessoal da Qualidade

- Um dos objetivos do PSP é ajudar as pessoas a aprender desde cedo como tratar de maneira realista e objetiva os defeitos no programa que resultam dos seus erros
- Mesmo sabendo que não somos infalíveis (cometemos erros): é difícil se sentir à vontade e tratar os problemas de maneira objetiva
- Maior parte dos problemas: erros datilográficos, distrações ou erros idiotas (maior sentimento de culpa)
 - para melhorar a situação não basta esforçar-se mais
 - é preciso enfrentar os problemas de maneira sistemática



PSP2: Processo de Gestão Pessoal da Qualidade

- Para gerenciar os erros é preciso conhecer os números
 - o PSP2 introduz as técnicas de inspeção e revisão para auxiliar na detecção precoce de defeitos
 - coleta e análise de dados de defeitos de compilação e teste encontrados em programas anteriores
 - é possível fazer listas de verificação mais ajustadas ao perfil de defeitos do programador e fazer avaliações da evolução seu nível de qualidade



PSP2.1: Processo de Gestão Pessoal da Qualidade

- Processo de design
- Auxilia no estabelecimento de critérios de completeza e de técnicas de verificação e consistência
 - critérios de completeza podem ser usados em qualquer fase, em requisitos, por exemplo
- São especialmente importantes para verificar se as condições para iniciar a próxima fase de desenvolvimento estão satisfeitas



PSP3: Processo Pessoal Cíclico

- Nos níveis de zero a dois: processo simples e linear para construir pequenos programas
- Abordagem não apropriada para projetos de maior envergadura
 - um programa com 10.000 LOCs é muito grande para ser escrito, depurado e revisto usando PSP2
 - mesmo bem documentado o programador facilmente perderá a noção da lógica geral do programa e perderá muito tempo tentando testar o programa todo como uma simples unidade
- PSP3 introduz abstração: subdividir o programa em módulos que possam ser tratados convenientemente com o ferramental apresentado nos níveis inferiores



PSP3: Processo Pessoal Cíclico

- Desenvolver iterativamente o programa, de módulo em módulo.
- Em cada iteração, existe um ciclo completo de design, codificação e teste, como no PSP2
 - pode ser aplicado para programas de vários milhares de linhas de código
- Fundamental que cada iteração tenha a sua qualidade controlada
 - em cada iteração, concentra-se na verificação da qualidade daquela iteração e assume-se que as anteriores já estão garantidas ou verificadas
 - se uma iteração anterior tem baixa qualidade, o teste será muito mais difícil e os benefícios do desenvolvimento incremental serão perdidos



PSP3: Processo Pessoal Cíclico

- Motivo para a ênfase em revisões e inspeções de design e código dos níveis inferiores de PSP: processo iterativo
- Importante:
 - relatórios
 - testes de regressão, para verificar se as condições de qualidade verificadas na iteração anterior não foram afetadas negativamente pela inclusão de novos módulos



Uso do PSP: expectativas

- Os desenvolvedores de software entenderão melhor o que eles fazem se o seu trabalho for definido, medido e acompanhado
- Eles usarão então uma estrutura definida de processos e critérios para avaliar e aprender com sua própria experiência e com a de outros
- Com este conhecimento e experiência, eles podem selecionar métodos e técnicas que melhor se ajustam aos tipos de tarefas que executam e às suas próprias habilidades
- Ao utilizarem práticas bem definidas e níveis elevados de qualidade no seu trabalho eles tornar-se-ão membros mais produtivos e eficazes nas equipes de desenvolvimento



Qualidade e produtividade: cinco princípios

- Um processo bem definido e estruturado pode melhorar a eficiência no trabalho
- O processo pessoal deve ser ajustado ao conhecimento e preferências de cada indivíduo
- Para que um desenvolvedor se sinta à vontade com um processo ele deve participar na sua definição
- Na medida em que o conhecimento e habilidade de um profissional evolui o mesmo deve acontecer com o processo utilizado
- A melhoria contínua fica facilitada com um processo de realimentação permanente



Efeitos do PSP

- Expectativa em atividades repetitivas e mecânicas:
 - efeito positivo da definição de processo sobre a produtividade
- Divergência:
 - efeito quando a atividade tem algum conteúdo de criatividade, como é o caso do desenvolvimento de software
- Dados em [Humphrey 95] são limitados:
 - desempenho de um pequeno número de estudantes ao longo de um treinamento em PSP



Resultados experimentais [Humphrey 95]

- Variação de produtividade na medida em que os estudantes se tornavam proficientes em PSP foi diminuindo
 - semelhante ao previsto no nível 2 CMM
- Redução na variação no caso do PSP pode ser creditada à melhoria da qualidade do código
- O número de defeitos por KLOC diminuiu ao longo do curso
 - diminuiu a necessidade de retrabalho devido à ocorrência de defeitos
 - indicador não foi medido
 - (autor comenta que indicadores de produtividade perdem sentido, devido à sua alta variabilidade, quando as taxas de ocorrência de defeitos é muito alta)



Resultados experimentais [Hayes 97]

- Dados experimentais coletados com 298 engenheiros de software:
 - a precisão nas estimativas de esforço melhorou 1,75 x
 - a precisão nas estimativas de tamanho de código melhorou 2,5 x
 - o número de defeitos encontrados no teste de unidade foi reduzido por um fator de 2,5
 - a porcentagem de defeitos encontrados antes da compilação aumentou em 50%
 - a produtividade pessoal em linhas de código por hora (na fase de codificação) não mudou de forma significativa
 - mas os autores acreditam que as melhorias em qualidade terão um impacto positivo quando forem considerados o ciclo de desenvolvimento como um todo, incluindo as atividades de integração e teste de sistema



PSP: Publicações

- [Humphrey 95]:
 - introduz o conceito do PSP e apresenta o método com um nível razoável de detalhe
 - contem a sequência apropriada de exercícios para a evolução nos quatro níveis de maturidade no nível individual
- [Humphrey 97]:
 - publicado posteriormente apresenta uma versão simplificada do método, uma introdução de caráter mais geral, sem o conceito dos quatro níveis, sem tratamento estatístico e sem os conhecimentos necessários para tratar programas de maior porte
 - o autor sugere que este livro seja usado como introdução para em seguida aprofundar-se na outra publicação



PSP: Conclusão

- Processo:
 - CMM no nível organizacional \Rightarrow PSP no nível do programador
- Nova abordagem do Prof. Humphrey:
 - TSP, Team Software Process: entre o nível individual do PSP e o nível corporativo do CMM
- Não foi proposto formalmente na literatura
 - tutorial sobre TSP programado para o SEI Software Engineering Symposium, em Pittsburgh, em setembro de 1998