

Dimensões a considerar na análise dos problemas de ensino e aprendizagem de engenharia de software

Cecília Keiko Adati Tomomitsu
Faculdade de Tecnologia de São Paulo - ckt_tomodati@yahoo.com.br

Vera Lúcia Silva Camargo
Faculdade de Tecnologia de Ourinhos - vlscamargo@terra.com.br

Aristides Novelli Filho
Faculdade de Tecnologia de São Paulo - anfmeq@uol.com.br

Resumo

Este artigo propõe quatro dimensões para a análise dos problemas de ensino e aprendizagem de Engenharia de software: Estrutura Curricular, Gestão de Currículo, Ensino e Aprendizagem.

A Dimensão Estrutura Curricular concentra-se em questões e conceitos filosóficos, epistemológicos e metodológicos de planos de cursos, como: a orientação a problemas, o pensamento reflexivo e a inclusão de pesquisa e experimentação.

A Dimensão de Gestão de currículo apresenta duas possíveis linhas de pesquisa: uma pedagógica, ampliação da inter e da transdisciplinaridade nos Projetos de Cursos, e outra tecnológica, criação de modelos de maturidade adaptados à cultura acadêmica que considerem a gestão do processo de ensino e aprendizagem em Engenharia de software.

Na Dimensão Ensino sugere-se pesquisar a utilização de Fábricas de Software aplicadas ao ambiente acadêmico e o estudo de como se aprende processos de software por meio da prática e experimentação.

Na Dimensão Aprendizagem pode-se estudar as motivações dos adultos para o aprendizado, a aplicação de métricas de qualidade de software para verificar como e quanto se aprendeu e quais processos de software seriam facilitadores da aprendizagem.

Essas quatro dimensões fundamentam o desenvolvimento das atividades de pesquisa dos autores e servem de guias na busca de alternativas de soluções que apoiem alunos e professores na produção de software de qualidade e transformem as Faculdades de Tecnologia – FATEC, com cursos na área de Tecnologia da Informação - TI, em Centros de Excelência em Engenharia de software.

Palavras-chave: Ensino, Aprendizagem, Engenharia de software, Estrutura Curricular, Gestão de Currículo.

1. Introdução

Nesses anos de atividades docentes os autores deste artigo têm percebido inúmeros problemas que afetam a aprendizagem de Engenharia de software adequada às necessidades das organizações. Isso os incentiva a pesquisar a taxionomia do processo de ensino-aprendizagem para obter respostas inerentes à complexidade do tema. A hipótese é que com a categorização serão facilitados o estudo, a compreensão e a elaboração de propostas que contribuam para a melhoria da qualidade do ensino e aprendizagem da Engenharia de software.

2. Objetivos

Este artigo propõe-se a focalizar a taxionomia do Processo de Ensino e Aprendizagem de Engenharia de Software, por meio da análise das dimensões da Estrutura Curricular, da Gestão de Currículo, do Ensino e da Aprendizagem, para contribuir com a comunidade científica e identificar linhas de pesquisa na área.

Essas quatro dimensões fundamentam o desenvolvimento das atividades de pesquisa dos autores e servem de guias para a definição das alternativas de solução aos problemas para apoiar alunos e professores na produção de software de qualidade e transformar as Faculdades de Tecnologia – FATEC, com cursos de Tecnologia da Informação - TI, em Centros de Excelência na área.

3. Discussão

Para iniciar essa discussão é apresentado, inicialmente, o conceito de Engenharia de software e em seguida analisa-se cada dimensão, Estrutura curricular, Gestão do currículo, Ensino e Aprendizagem, com relação aos problemas de ensino e aprendizagem de engenharia de software e às suas possibilidades de linhas de pesquisa.

3.1 Conceito de Engenharia de Software

Engenharia de Software - ES é o conjunto organizado de métodos, ferramentas e técnicas, para produzir software com qualidade. Meyer [1] diferencia a simples programação de computadores da Engenharia de Software pela ênfase que se dá nesta última à questão da qualidade e pelo seu caráter industrial.

3.2. Dimensão - Estrutura curricular

A Dimensão Estrutura Curricular concentra-se em questões e conceitos filosóficos, epistemológicos e metodológicos de planos de cursos, como: a orientação a problemas, o pensamento reflexivo e a inclusão de pesquisa e experimentação.

São diversas questões a tratar nos currículos de cursos da área. Têm-se problemas de falta de integração das disciplinas, de desequilíbrio entre teoria e

prática. No âmbito dos cursos de Tecnologia, os alunos questionam conteúdos das disciplinas de apoio como muito teóricos. O profissional de ES necessita mapear as regras de negócio da organização em uma solução de software e percebe-se a falta de conhecimentos práticos sobre o *modus-operandi* das organizações, o que talvez possa ser resolvido com mais atividades de experimentação no desenvolvimento do curso.

Bordenave [2] denomina as vivências que influenciam a formação de currículo invisível. Para ele, quando o currículo é organizado como uma colcha de retalhos, com disciplinas estanques, sem integração orgânica e funcional, o profissional formado reflete a falta de integração do conhecimento e adquire uma visão fragmentada da realidade, pois essa é a vivência disponibilizada pelo currículo invisível. Engenharia de Software no currículo dos cursos de TI é um conjunto de disciplinas, da programação de computadores à metodologia de desenvolvimento e implementação de sistemas. Parte-se da disciplina mais particularizada para a de maior abstração, abordagem "*botton-up*". A fragmentação e a falta de um processo de ES integrador das disciplinas específicas impedem que o aluno tenha uma visão real da ES praticada nas organizações e um treino da abordagem "*top-down*", partir do mais abstrato para depois definir os detalhes.

No Brasil, diferentemente de outros países, não há uma formação de graduação apenas em Engenharia de software. Nos Estados Unidos, Meyer [1] define cinco objetivos para um currículo exclusivo de ES: princípios, boas práticas de ES, aplicações, ferramentas e matemática como base formal. Destaca também as competências que o profissional deve desenvolver ao longo do curso, principalmente a capacidade de abstração.

3.3. Dimensão - Gestão do currículo

A Dimensão de Gestão de currículo apresenta duas possíveis linhas de pesquisa. Uma linha pedagógica de estudos, ampliação da inter e da transdisciplinaridade nos Projetos de Cursos, e outra tecnológica, criação de modelos de maturidade adaptados à cultura acadêmica que considerem a gestão do processo de ensino e aprendizagem em Engenharia de software.

Para uma gestão de curso mais eficiente e eficaz, o projeto pedagógico do curso deveria prever atividades facilitadoras da inter e transdisciplinaridade.

Interdisciplinaridade é o esforço entre os docentes na comunicação vertical entre semestres seqüenciais ou adjacentes, para garantir a evolução encadeada do conhecimento como produto do processo de ensino, vista como um processo externo ao aluno, focado no conteúdo.

Transdisciplinaridade reflete a transcendência do relacionamento entre professores e alunos, além do diálogo interdisciplinar, para substituir a relação hierarquizada entre docência e discência pela relação cooperativa. Atua na forma como o processo de comunicação de conteúdos se dá, por meio do professor, para otimizar a participação, comprometimento e aprendizagem do aluno, vista como um processo internalizado, focado no aluno e destinado ao desenvolvimento do trabalho que alinhe processo e assunto [3].

Atividades interdisciplinares podem ser desenvolvidas na Fábrica de Software para permitir além da Gestão de planos de Ensino (conteúdos), a Gestão dos Projetos (estruturas) que interligam os conhecimentos.

Os gestores deveriam aplicar os mesmos instrumentais ensinados aos alunos, a gestão de processos como o *Business Process Management – BPM* e os modelos de maturidade, como o *Capability Maturity Model – CMM*.

Poder-se-ia também contar com um repositório de softwares desenvolvidos ao longo dos anos, com dados históricos, assim como, métricas de processo de software da Faculdade para as melhorias constantes no processo de Gestão do próprio currículo.

Métricas são medidas úteis para se obter efetividade de um determinado processo de software e a qualidade do produto. Conn [4] tem uma proposta de métricas adaptadas à cultura acadêmica, aplicáveis em cursos da área e com uso do processo de software, *Team Software Process – TSP*, do *Software Engineering Institute – SEI*. Ele estabelece, dois processos, o das aulas e o do projeto de software. Processo das Aulas – PA é o da transmissão de conteúdos compostos pelo conjunto de atividades que direcionam as demais e que analisa o processo de software utilizado pelos alunos para resolver os problemas do projeto. Processo de Engenharia de Software – PES referente ao projeto desenvolvido pelos alunos em paralelo com as aulas.

3.4 Dimensão - Ensino

Na Dimensão Ensino sugere-se pesquisar a utilização de Fábricas de Software aplicadas ao ambiente acadêmico e o estudo de como se aprende processos de software por meio da prática e experimentação.

Uma das possibilidades de melhorar o aprendizado de processos de software é a Fábrica Acadêmica de Software - FAS, um ambiente real de ES para os docentes e alunos realizarem pesquisa de processos, métodos e ferramentas e criarem softwares de qualidade, como proposto por Tvedt [5].

Fábrica de Software é uma solução tecnológica como linha de montagem, à semelhança dos processos de produção da indústria tradicional, composta por processos padronizados, para se obter qualidade e escala. Nela são executadas todas as fases de desenvolvimento de sistemas, da especificação à implantação, o que compreende dois grandes processos: o de engenharia de processos e o de engenharia de software [6].

Fábrica de Software como ambiente de experimentação de projetos, experimentação essa necessária em Engenharia de software, que não pode abrir mão da prática, serviria também para se definir uma epistemologia própria em Ensino de processos de software como propõe Shön [7].

3.5. Dimensão - Aprendizagem

Na Dimensão Aprendizagem pode-se estudar as motivações dos adultos para o aprendizado, a aplicação de métricas de qualidade de software para verificar como e quanto se aprendeu e quais processos de software seriam facilitadores da aprendizagem.

Questões relativas à aprendizagem devem ser abordadas associadas àquelas relacionadas ao Ensino, uma vez que o alvo do ensino é o aprendiz, e no contexto dos cursos de graduação, um aprendiz adulto. Reconhece-se que as estratégias e abordagens aplicadas ao ensinar deveriam nortear-se para a

facilitação do pleno aprendizado. Esse é o grande desafio, determinar quais estratégias e abordagens devem ser lançadas para que o ensino seja focado para o aprendizado. Abreu [8] destaca que o papel do professor desponta como o de facilitador da aprendizagem de seus alunos. Seu papel não é ensinar, mas ajudar o aluno a aprender.

Frente a essas considerações depara-se com posturas e estratégias de ensino-aprendizagem, de ES, nas quais os docentes utilizam estudos de caso e problematização em sala de aula, e os alunos, em equipe, desenvolvem os softwares para resolver os problemas do caso estudado, pequenos projetos com ciclo de desenvolvimento de um a dois semestres, com dedicação parcial por parte dos alunos, quando na vida real os projetos relevantes para aplicação das metodologias exigem tempo integral de dedicação, são de ciclos longos e muito mais complexos do que o projeto escolar. A situação é artificial, pois o aluno não desenvolve o software, em um ambiente real de pressão pela qualidade, custos e prazos, e com isso não compreendem a relevância do processo de software e têm poucas oportunidades de conhecê-lo tal como ocorre nas organizações. Após a avaliação dos softwares desenvolvidos pelos alunos, os mesmos são descartados, e nas turmas seguintes, outros projetos são desenvolvidos. Não há sequer um aproveitamento desses softwares em um repositório para aplicar dois princípios primordiais da Engenharia de software, o reuso e a criação de métricas.

No processo de aprendizagem o elemento referencial é o “aprendiz”, esta acontece no aprendiz e pela pessoa do aprendiz. Ele é o principal envolvido neste processo e tem claros os seus objetivos: aprender, “o quê”, e “para quê”. O reconhecimento pelo aprendiz da importância daquilo que deve aprender torna o processo de aprendizagem significativo. (ABREU [8]).

O aprendiz deve explicitar o “querer” aprender. A demanda por cursos das FATECs e as análises conhecidas sobre a situação dos formados no mercado de trabalho permite intuir este “querer”.

O aprendiz, centro do processo da aprendizagem, Ser Humano que evolui e desenvolve a sua maturidade ao longo da sua vivência, consequência das suas experiências em vida, adquire a capacidade de aprender ao longo dessa vivência. O “quando” é outra importante variável a influenciar a aprendizagem, e essa variável é diretamente dependente do aprendiz. Cabe a ele identificar a melhor forma de aprender, o “como” aprender e fazer uso das melhores técnicas e habilidades para obter o máximo desse processo de aprendizagem.

Para garantir a eficiência e eficácia da aprendizagem, “quanto” o aprendiz aprendeu, deve-se medir e verificar a qualidade do processo de aprendizagem: o “saber” (conhecimento) e o “saber fazer” (habilidade). Esse “quanto” é evidenciado pela demonstração dos conhecimentos e habilidades desenvolvidos e permite concluir que o objetivo foi ou não alcançado.

A garantia da qualidade da aprendizagem deve, de acordo com o objetivo estabelecido, apoiar-se na avaliação do processo e do aprendiz. A avaliação adequada oferece informações relevantes que podem, por meio de indicadores previamente criados, mensurar a qualidade da aprendizagem.

Qualidade é a totalidade de características de uma entidade que lhe confere a capacidade de satisfazer necessidades explícitas e implícitas. Para a garantia da qualidade do produto de software dos aprendizes poder-se-ia aplicar um conjunto de atividades planejadas e sistemáticas adequadas ao ambiente de ensino de ES para aumentar a confiança de que o resultado final atenderá requisitos de qualidade.

Essa avaliação da qualidade do produto de software deve contemplar: Requisitos Funcionais; Requisitos Não-funcionais e Requisitos de Usabilidade. (NBR ISO 8402 [9]; NM-ISO 8402 [10]).

Poder-se-ia utilizar as métricas para qualidade de processo e produto de software para avaliar aprendizagem, conforme proposta de Conn (2004). Para ele o projeto de software, como elemento de aprendizagem, é composto por códigos, requisitos, projeto das especificações, planos de gestão e os dados coletados para as métricas com o registro das horas trabalhadas, contagem dos defeitos por tipo e a contagem do número de classes, objetos, atributos e linhas de código. Com dados coletados derivam-se medidas tais como duração de cada subprocesso, densidade de defeitos (número de defeitos por objeto, linha de código ou por linha de texto), e para cada produto de software determina-se a produtividade da equipe e a individual dentro da equipe, assim como a eficiência (número de linhas de texto, número de linhas de código e número de objetos por hora). Nessa proposta os estudantes representam papéis típicos das equipes de projeto, como membro ou líder de equipe, gerente de desenvolvimento, de planejamento, de qualidade ou de suporte.

Além das métricas deve-se trabalhar com processos de software mais adequados ao tempo acadêmico, como os métodos ágeis. *Extremme Programming-XP*, *Crystal Clear*, *Scrum* são exemplos desses métodos que permitem uma apropriação de conhecimento pelo aluno mais rápido do que as metodologias tradicionais, ou as mais extensas como o *Unified Software Development Process – USDP*.

4. Conclusões

Embora as quatro dimensões (Estrutura curricular, Gestão do currículo, Ensino e Aprendizagem) sejam relacionadas, pois uma dimensão interfere na outra, subdividir e categorizar os problemas deixa mais compreensível quais as possibilidades de pesquisas.

Poucos são os autores que fundamentam suas propostas de currículos com o referencial teórico da área de Educação. Essa é uma direção importante a ser tomada, pois as abordagens filosóficas, epistemológicas e educacionais dos currículos e respectivos planos de cursos, assim como, a discussão da orientação a problemas, a questão do pensamento reflexivo nessa área e a necessidade da experimentação e da pesquisa de aplicativos podem contribuir para uma formação de mais qualidade dos profissionais de ES.

Gestão de processos e as métricas de qualidade de software adaptadas à cultura acadêmica não são utilizadas nesse contexto, portanto definir os fundamentos que possam favorecer a gestão, tanto de currículos quanto de processos de ensino, é uma atividade relevante.

Ensinar focado no aprendiz é fundamental para desenvolvimento das competências profissionais. O detalhamento de uma epistemologia da prática de Engenharia de software e de estudos que aumentem a significância desse Ensino para os alunos torna-se linha de investigação a ser aprofundada.

Os métodos, técnicas e ferramentas da Engenharia de software, também podem ser aplicados na avaliação da aprendizagem. As métricas de qualidade de software devem ser utilizadas para verificação de quanto e como se aprendeu.

Processos de software facilitadores da aprendizagem é outra possibilidade de pesquisa.

Essa categorização dos problemas de Ensino e aprendizagem de Engenharia de software em quatro dimensões apoiará os autores do artigo em suas pesquisas futuras.

5. Referências

- [1] MEYER, Bertrand. **Software Engineering in the Academy**. Computer 34, 5 (May. 2001), 28-35. IEEE Computer Society Press, 2001.
- [2] BORDENAVE, Juan Díaz. **A pedagogia da problematização na formação dos profissionais de saúde**. In: Fórum de Brasília em Saúde Educação. UFMS, 1996. Disponível em: www.ead.ufms.br/ambiente/progesp/saude-educacao/PEDAGOGIA_PROBLEMATIZADORA.doc. Acesso em: 10 de jul. de 2006.
- [3] LEITE, Denise. **Conhecimento Social na Sala de Aula Universitária e a Autoformação Docente**; In: Professor do Ensino Superior - Identidade, Docência e Formação (Marília C. Morosini - org.). Planalto, 2001.
- [4] CONN, Richard. **A reusable, academic-strength, metrics-based software engineering process for capstone courses and projects**. In: Proceedings of the 35th SIGCSE technical symposium on Computer science education. p.492 – 496. ACM Press, New York, 2004.
- [5] TVEDT, John D.; TESORIERO, Roseanne, and GARY, Kevin A. **The software factory: combining undergraduate computer science and software engineering education**. In: Proceedings of the 23rd international Conference on Software Engineering (Toronto, Ontario, Canada, May 12 - 19, 2001). International Conference on Software Engineering. IEEE Computer Society, Washington, DC, 633-642.
- [6] CUSUMANO, M. A. **The Software Factory: A Historical Interpretation**. IEEE Software, March 1989, pp. 23-30.
- [7] SHÖN, Donald A. **Educando o profissional reflexivo: um novo design para o ensino e a aprendizagem**. Trad. Roberto Cataldo Costa. Porto Alegre: Artes Médicas, 2000.
- [8] ABREU, Maria Célia de; MASETTO, Marcos T. **O professor Universitário em aula: prática e princípios teóricos**. São Paulo: MG Ed. Associados, 1990.
- [9] ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. NBR ISO 8402: gestão da qualidade e garantia da qualidade. Rio de Janeiro. 1994.
- [10] COMITÉ MERCOSUR DE NORMALIZACIÓN. NM-ISO 8402:97: **Gestión de la calidad y aseguramiento de la calidad** – Vocabulario.1997.

Contactos

Cecília Keiko Adati Tomomitsu (ckt_tomodati@yahoo.com.br)

Professora da Faculdade de Tecnologia de São Paulo

Departamento de Processamento de Dados

Praça Cel. Fernando Prestes, 30 - Bom Retiro – São Paulo

CEP: 01124-060

Celular: 7127-9699

Vera Lúcia Silva Camargo (vlscamargo@terra.com.br)

Professora das Faculdades de Tecnologia de Ourinhos e Carapicuíba

Rua Abuassali Abujamra, 519 – Jd. Aurora – Ourinhos –SP

CEP: 19907-170

Celular: (14) 9621 0146