Investigação da Rotina de Aprendizagem de Desenvolvedores de Software

Geovane F. Piccinin, Mirella M. Moro

Universidade Federal de Minas Gerais (UFMG), Belo Horizonte – MG – Brazil

{geovanepcc,mirella}@dcc.ufmg.br

Abstract. *This work describes a qualitative research carried out with professional software developers regarding their learning routines. We investigate participants' perceptions about their learning tasks and strategies used. We consider an interview script based on the Self-Regulation of Learning model. The results indicate the ability to learn is crucial for such professionals and their routines are very similar to that of self-regulated learners.*

Resumo. *Este trabalho descreve uma pesquisa qualitativa realizada com profissionais desenvolvedores de software a respeito de suas rotinas de aprendizagem. Investigamos as percepções dos participantes sobre suas tarefas de aprendizagem e estratégias empregadas. Utilizamos um roteiro de entrevistas baseado no modelo de Autorregulação da Aprendizagem (Self-Regulated Learning). Os resultados indicam que a capacidade de aprender é crucial para esses profissionais e que suas rotinas muito se assemelham a de aprendizes autorregulados.*

1. Introdução

A área de desenvolvimento de software exige que seus profissionais estejam em constante aprendizagem, pois tecnologias existentes são atualizadas e novas surgem regularmente. Tal realidade faz com que desenvolvedores estejam sempre buscando informações atuais (novas) para a solução de problemas na execução de seu trabalho. Dessa forma, é esperado que desenvolvedores de software possuam certas habilidades que se assemelham àquelas definidas para aprendizes autorregulados. O aprendiz autorregulado planeja, monitora, avalia e ajusta seu comportamento e estratégias para atingir seus objetivos [Schunk and Zimmerman 2011]. A autorregulação da aprendizagem (*Self-Regulated Learning* ou SRL) vem sendo estudada no ambiente acadêmico e profissional [Broadbent and Poon 2015, Littlejohn et al. 2016, Milligan and Littlejohn 2016, Schunk and Zimmerman 2011], inclusive no contexto de aprendizagem de computação [Falkner et al. 2014, Pedrosa et al. 2016]. Porém, faltam estudos direcionados para o cenário de desenvolvimento profissional de software, que é especialmente peculiar devido ao seu altíssimo grau de inovação.

O objetivo deste estudo é avaliar inicialmente a presença dos aspectos da SRL no âmbito do desenvolvimento profissional de software, assim como identificar as estratégias de SRL empregadas pelos profissionais. Para tal, realizamos entrevistas semiestruturadas com 10 profissionais da área de perfis heterogêneos. Por fim, discutimos a visão atual do suporte a SRL e apontamos a necessidade de mais esforços de pesquisa na área.

2. Trabalhos Relacionados

A autorregulação da aprendizagem é o direcionamento dos esforços cognitivos, ações e emoções que são planejados e adaptados ciclicamente para a obtenção de objetivos de aprendizagem [Schunk and Zimmerman 2011]. Existem diversos modelos que oferecem diferentes perspectivas sobre SRL [Panadero 2017], entre elas, destaca-se o modelo proposto Schunk e Zimmerman [2011]. Ele é composto pelas fases de planejamento, desempenho e reflexão, e considera um processo iterativo que acontece constantemente durante a aprendizagem. A partir desses diferentes modelos, também existem instrumentos para mensurar a SRL, como discutido em [Rovers et al. 2019]. Por exemplo, [Fontana et al. 2015] introduz um instrumento criado para mensurar SRL no ambiente de trabalho. No caso de aplicação de estratégias em contexto de desenvolvimento de software, destacam-se trabalhos sobre autorregulação presente nas atividades de aprendizagem de programação no ambiente acadêmico [Çakiroglu et al. 2018, Falkner et al. 2014, Loksa and Ko 2016, Pedrosa et al. 2016].

Aqui, utiliza-se como referência o modelo de Schunk e Zimmerman [2011], por ser o precursor no qual outros são propostos com atualizações que não apresentam vantagens específicas para este estudo. As perguntas do roteiro de entrevista basearam-se nas fases propostas por tal modelo, mas não se contiveram a nenhum questionário prévio. Essa medida foi adotada por conta da natureza mais exploratória desta pesquisa. Finalmente, este estudo avança tal estado-da-arte por apontar a demanda de profissionais desenvolvedores de software por estratégias de autorregulação da aprendizagem e que estas devem ser específicas às particularidades da carreira, conforme descrito a seguir.

3. Breve Descrição da Metodologia

Foi realizada uma pesquisa qualitativa baseada em entrevista semiestruturada com 10 participantes no período de Março a Abril de 2020. Os perfis dos participantes incluem: graduados em Sistemas de Informação ou Ciência da Computação, sendo quatro mestres em Ciência da Computação, sete homens e três mulheres, que atuam profissionalmente com desenvolvimento de software em sete empresas distintas, formando uma amostra heterogênea. A experiência de trabalho dos participantes varia de um ano e meio até 20 anos. Desconsiderando esses *outliers*, a média de tempo de experiência é de seis anos. O recrutamento dos participantes foi através de contato pessoal e indicação, e as entrevistas foram realizadas por telefone com duração de cerca de 30 minutos. Essa metodologia foi escolhida por se tratar de uma investigação inicial a fim de identificar estratégias particulares dos participantes, mais do que determinar se eles são autorregulados ou qual tipo de estratégia mais utilizada. O roteiro de entrevista considera blocos temáticos: (1) dados de perfil; (2) motivação; (3) rotina de aprendizagem; (4) planejamento; (5) estratégias de aprendizagem; (6) avaliação; (7) reflexão; e (8) uso de ferramentas de apoio a aprendizagem. Os blocos temáticos (4), (5), (6) e (7) buscaram compreender as fases definidas no modelo de SRL adotado como referência [Schunk and Zimmerman 2011].

4. Resultados e Discussão

Embora o modelo de SRL utilizado ofereça um vocabulário de categorias, a análise das transcrições foi feita utilizando uma codificação aberta, a qual extrai informações úteis das transcrições e pontos que se destacaram na fala dos participantes. Depois, na codificação

axial, houve um realinhamento das categorias que se conciliavam com as encontradas no modelo de SRL quando possível. Um processo parecido executado com êxito é descrito em [Falkner et al. 2014]. A seguir, apresentamos e discutimos os resultados com foco nas quatro perguntas centrais ao modelo SRL (i.e., 4 a 7 conforme Seção 3).

Rotina de Aprendizagem e Impacto na Carreira. Todos os participantes confirmaram a ideia inicial de que a carreira de desenvolvedor de software requer constante aprendizagem. Essa afirmação parece meio óbvia e é algo que professores/pesquisadores estão acostumados na Academia; mesmo assim, é importante averiguar tal noção com participantes devido à metodologia SRL utilizada. Exemplos de respostas para tal questão incluem as seguintes (participantes anonimizados por números).

P5: "*Eu tenho que pesquisar a todo momento, sempre tem alguma coisa que eu não sei fazer.*" **P8**: "*Aprender é uma parte crucial da profissão de desenvolvedor... todo dia você tem de aprender uma coisa nova, é um processo que nunca vai acabar até o fim da sua carreira.*" **P9**: "*Eu considero que meu trabalho seja o trabalho de aprender a resolver problemas. No fundo, eu sou pago para aprender aquilo que é necessário para resolver um problema, mais do que já saber alguma coisa de antemão.*"

Planejamento e Estratégias de Aprendizagem. De acordo com os relatos, a busca por informação acontece somente conforme o necessário. Uma hipótese para explicar é que como as tecnologias são muito variadas e podem mudar entre projetos, não faz sentido aprender para além da demanda. Nas questões relativas a planejamento, gerenciamento do tempo e horário de estudos, respostas relevantes incluem as seguintes.

P2: "*Eu não faço nenhum planejamento. Eu vou aprendendo por demanda no meu horário livre.*" **P4**: "*Eu não consigo dizer se aprendo mais no meu horário de trabalho ou fora dele. Geralmente, quando tem entrega, eu tenho que usar meu tempo livre para estudar.*" **P10**: "*Eu defino com meu supervisor uma espécie de escada, vou pegando tarefas mais simples primeiro e depois as mais difíceis.*" **P7**: "*Eu não tenho planejamento. O que eu olho é assim: vou querer aprender até tanto hoje, eu fico naquilo ali até acabar aquele assunto, independente do número de horas.*"

Os participantes, assim como aprendizes autorregulados, mostraram conhecer as estratégias de aprendizagem que os favorecem, conforme a seguir.

P1: "*Primeiro eu olho tutoriais daquela tecnologia, leio e depois eu tento repetir o que tá nele. Eu prefiro material escrito, raramente vejo vídeos.*" **P10**: "*Busco indicação de pessoas sobre os materiais. Se a documentação for boa eu vou na documentação, senão, procuro por um curso que tiver bom review.*" **P5**: "*Geralmente eu pego um código pronto e vou modificando ele. Gosto de ler o código dos outros e ir variando para ver o que acontece.*" **P4**: "*Nos fóruns eu vou mais para resolver problemas. Às vezes eu posto lá, mas prefiro mais a documentação.*"

Nesses depoimentos, foi possível notar dois aspectos fundamentais: o pragmatismo das estratégias de aprendizagem e a diferença de estilos de aprendizagens. Existem argumentos distintos para as diferentes opções de mídias, e isso pode acontecer devido à diferença de estilos de aprendizagem [Çakiroglu et al. 2018, de Oliveira et al. 2018]. Quanto ao uso de materiais complementares como anotações e diagramas, apenas dois afirmaram fazer anotações.

Quanto a cursos formais, dois participantes afirmaram a opção por cursos breves justificando-se pela busca de credibilidade da informação e porque o curso é em si uma espécie de curadoria de conteúdo. Outros demonstraram oposição a cursos, pois preferem aprender no próprio ritmo e não no imposto pelo mesmo.

A busca de contato com outras pessoas para o aprendizado foi quase sempre apontada como último recurso. Em geral, os participantes falaram que aprender era um processo que faziam sozinhos e só buscam outras pessoas em último caso. Não ficou claro o motivo, e essa questão pode ser melhor investigada em investigações subsequentes.

Avaliação das Estratégias e Dificuldades na Rotina de Aprendizagem. Os participantes afirmaram que a forma de avaliar se estão atingindo seu objetivo de aprendizagem é verificar se estão conseguindo realizar as tarefas na prática. No entanto, apenas fazer funcionar não foi considerado como prova de que se aprendeu. Em todos os casos, o código que funciona e que é bem entendido apresentou-se como o verdadeiro registro da aprendizagem, que é inclusive usado posteriormente para consulta pelo próprio aprendiz. Nota-se também que, ao perceberem que não estão aprendendo, a estratégia então é refazer o percurso de aprendizagem; às vezes, buscando por um novo material ou material de outro formato. Novamente, a busca por contato direto com pessoas foi citada como o último recurso, conforme destacado nas respostas a seguir.

P2: "*Eu não tenho nenhuma forma assim de dizer se eu estou aprendendo. Se eu não tiver conseguindo fazer a tarefa eu tenho que estudar, ler de novo.*" **P8**: "*Quando eu consigo fazer alguma coisa do zero é que eu considero que estou aprendendo. Ultimamente eu uso essa estratégia. Antes eu seguia um tutorial e ficava com a ilusão que tinha aprendido. Por isso tem que fazer parecido com o tutorial, mas diferente e sem ajuda.*" **P1**: "*Se eu fizer a tarefa e entender o código é sinal que eu aprendi, porque código dá para copiar e colar também, e às vezes fica funcionando mas sem ninguém saber o que tá acontecendo. Eu prefiro aprender, mas quando não tem tempo fica sem saber mesmo.*" **P3**: "*Na hora de implementar uma coisa eu procuro no código se alguém já fez algo parecido para eu entender. Procuro também o que eu já fiz.*"

Sobre as principais dificuldades relativas à tarefa de aprendizagem, a mais recorrente refer-se à qualidade da documentação, além de vocabulário novo e conhecimento prévio requerido, e disciplina, conforme a seguir.

P9: "*Tem framework com a documentação muito ruim, e aí você tem que ficar buscando e muitas vezes os tutoriais são muito superficiais.*" **P2**: "*A dificuldade é que para saber aquilo que está na documentação você tem que saber outras coisas antes.*" **P6**: "*Meu problema é mais de disciplina. Achar um horário para estudar, principalmente fora do serviço, em casa o ambiente tem muita distração.*"

Estratégias e Uso de Ferramentas de Apoio a Aprendizagem. A maioria afirmou considerar suas estratégias atuais suficientes, embora gostariam que fossem mais eficientes em relação à questão de planejamento e horários. A maioria dos participantes afirmou não usar nenhum tipo de ferramenta de apoio à aprendizagem e não apresentaram argumento que justificasse o porquê. Dentre os que afirmaram usar, o relato é do uso de editores de texto para registrar anotações.

5. Conclusão

A partir das entrevistas, ficam evidentes a importância de os profissionais gerenciarem o aprendizado de novas tecnologias, e a semelhança entre rotina de aprendizagem e perfil de aprendizes autorregulados, indicando que o apoio a SRL possa ser proveitoso para tais profissionais. Ainda sobre as respostas do questionário, a fase de planejamento apareceu como a etapa de SRL com maior carência e a que os participantes gostariam de tornar mais eficaz. Os resultados deixam claro que mais estudos sobre a SRL e a rotina de aprendizagem de desenvolvedores de software podem ajudar a reconhecer formas de melhorar o apoio à autorregulação desses profissionais, e será nosso foco futuro. Como limitações, entendemos que a amostra de participantes poderia ser maior para generalizar os resultados obtidos. Mesmo assim, tal amostra permitiu obter uma visão geral que pode guiar novos estudos.

Agradecimentos. Trabalho parcialmente financiado por FAPEMIG e CNPq. Autores agradecem aos participantes da entrevista.

Referências

Broadbent, J. and Poon, W. (2015). Self-regulated learning strategies & academic achievement in online higher education learning environments: A systematic review. *The Internet and Higher Education*, 27:1–13.

Çakiroglu, Ü. et al. (2018). Exploring the use of self-regulation strategies in programming with regard to learning styles. *Int'l J. of Computer Science Education in Schools*, 2(2).

de Oliveira, E. S., Sales, G. L., de Sousa Pereira, P., and do Nascimento Moreira, R. (2018). Identificaçao automática de estilos de aprendizagem: Uma revisao sistemática da literatura. In *Anais do XXVI Workshop sobre Educação em Computação*. SBC.

Falkner, K. et al. (2014). Identifying computer science self-regulated learning strategies. In *ITiCSE*, pages 291–296.

Fontana, R. et al. (2015). Measuring self-regulated learning in the workplace. *Int'l J of Training and Development*, 19(1):32–52.

Littlejohn, A. et al. (2016). Professional learning through everyday work: how finance professionals self-regulate their learning. *Vocations and Learning*, 9(2):207–226.

Loksa, D. and Ko, A. J. (2016). The role of self-regulation in programming problem solving process and success. In *ICER*, pages 83–91.

Milligan, C. and Littlejohn, A. (2016). How health professionals regulate their learning in massive open online courses. *The Internet and higher education*, 31:113–121.

Panadero, E. (2017). A review of self-regulated learning: six models and four directions for research. *Frontiers in psychology*, 8:422.

Pedrosa, D. et al. (2016). Self-regulated learning in computer programming: strategies students adopted during an assignment. In *iLRN*, pages 87–101.

Rovers, S. F. E. et al. (2019). Granularity matters: comparing different ways of measuring self-regulated learning. *Metacognition and Learning*, 14(1):1–19.

Schunk, D. H. and Zimmerman, B. (2011). *Handbook of self-regulation of learning and performance*. Taylor & Francis.