Self-Regulated Learning in Higher Education: Strategies Adopted by Computer Programming Students

Daniela Pedrosa[1, 2], José Cravino[1, 2], Leonel Morgado [3, 4], Carlos Barreira [5]

[1] Universidade de Trás-os-Montes e Alto Douro (UTAD), Vila Real, Portugal.
[2] Research Centre "Didactics and Technology in Education of Trainers", Aveiro, Portugal.
[3] Universidade Aberta, Coimbra, Portugal.
[4] INESC TEC, Porto, Portugal
[5]Faculdade de Psicologia e Ciências da Educação da Universidade de Coimbra, Coimbra, Portugal.

Email: dpedrosa@utad.pt, jcravino@utad.pt, leonel.morgado@uab.pt, cabarreira@fpce.uc.pt

Abstract

To help students overcome their learning difficulties in the transition from entry-level to advanced computer programming, developing an appropriate set of learning strategies, the SimProgramming teaching approach has been adopted at the University of Trás-os-Montes e Alto Douro (Portugal). This approach is based on four conceptual foundations: business-like learning environment, self-regulated learning, co-regulated learning, and formative assessment. In this approach the students develop an activity based on problem-based learning, with a specific set of tasks based on those four conceptual foundations.

The approach was implemented in two courses from the second and third curricular years of the bachelor programmes in Informatics Engineering and Information & Communication Technologies. We conducted semi-structured interviews with students (n=32) at the end of the courses, to try to identify the students' strategies for self-regulation of learning in the activity developed within the SimProgramming approach. The main strategies identified were: organization, planning, time management, identification of difficulties, resolution of the difficulties encountered, work review, identification of the factors that influenced their motivation, and structure of the environment.

The factors influencing the motivation most often identified by students were the impact of the assessment in the final course grade, the completion of the course, learning, skills development, and teamwork. Generally, students applied strategies to solve the difficulties, in particular by searching for social help and information search. Procrastination was also often identified by students. Strategies of time management, transformation of information, in-depth review, self-reflection, and self-evaluation were referenced scantily. We found that students changed some of their strategies from one course edition to the next.

We conclude by recommending the development of educational practices to help students review their work, treat and process the information they find, conduct self-reflection and self-evaluation of their performance during tasks, adopt concentration strategies, and become aware of their specific difficulties.

Keywords: Self-Regulation Learning; Strategies of Self-Regulation Learning; Computer Programming.

1 Introduction

A common trend in computer programming courses in higher education are the high rates of academic failure and students struggling, particularly in the transition from entry-level programming to advanced programming. Reasons pointed out in the literature include the teaching approach and the attitudes/strategies used by students in computer programming (Gomes & Mendes, 2007).

In higher education, self-regulated learning (SRL) is a key element, because it allows students to be proactive and manage their learning and development of life skills (Fernández et al., 2013). The application of SRL strategies typically predicts high academic achievement (Broadbent & Poon, 2015), and the self-regulatory processes can be improved with appropriate interventions (Zimmerman, 2008; Fernández, 2013). For example, it is recommended that teachers contribute to the development of metacognitive activities, of skills for implementing and adapting strategies for self-monitoring, make strategic use of feedback, and promote students' development of metacognitive knowledge about academic work and task-specific strategies (Cazan, 2013).

We applied the SimProgramming pedagogical approach (Pedrosa et al., 2016), in the academic years 2012/2013 and 2013/2014, in two intermediate programming courses from the second and third curricular years of the bachelor programmes in Informatics Engineering (IE) and Information & Communication Technologies (ICT). In the SimProgramming approach, the students develop a problem-based learning activity within the syllabus of the respective course, with a specific set of tasks based on the conceptual foundations detailed ahead.

We conducted semi-structured interviews with students at the end of the activity (n=32), to identify the self-regulation learning strategies that students employed, in two cycles: one in the Programming Methods 3 (PM3) course and another in the Programming Methods 4 (PM4) course.

2 Background

Computer programming courses are complex (Robins, Rountree & Rountree, 2003), students experience difficulties learning (Lahtinen, Ala-Mutka & Järvinen, 2005), and lack motivation and involvement in study (Morgado et al., 2012; Nunes et al., 2015). In intermediate and advanced programming courses, the level of complexity is much greater than entry-level programming courses. For example, students have difficulties grasping architectural styles such as Model–View–Controller (MVC) and other software engineering concepts (Cagiltay, 2007; Morgado et al., 2012), or dealing with the context of Web programming, where code is neither written nor executed in a single location, but rather distributed between the server and the client, and applications need to operate over the HTTP protocol which was designed to be stateless, hence encumbering applications with kludge-like solutions such as passing around session data in cookies or address parameters, among other difficulties (Liu & Phelps, 2011).

In learning computer programming, the students that apply SRL and metacognitive strategies have a good performance (Bergin, Reilly & Traynor, 2005). However, the most students in computer science are not aware of SRL and metacognitive strategies, and it is necessary to infuse them in this context (Alharbi et al., 2011).

SRL is considered a meta-process that depends on the active participation of students developing academic skills, in the selection of learning strategies when conducting an academic assignment (Clark, 2012). Students demonstrate their proactive competences, monitoring and adapting the learning processes, for the regulation of metacognitive, cognitive, motivational, behavioral and environmental strategies for achieving personal goals (Zimmerman & Schunk, 2007). The interaction between the compromise, self-control, autonomy and students' self-discipline allows regulating their actions to achieve their learning goals (Hattie & Timperley, 2007).

Self-regulated learners are characterized by constructing their own meanings, goals, and strategies from the information available in the external environment and in their own minds (Pintrich, 2004). Their level of domain of self-efficacy and self-knowledge, appeals to various learning strategies (Zimmerman, 2013), and the acquisition of effective practices for their study, such as: time management; resource management; environmental management; incorporating feedback; management of learning objectives and results (Nicol & Macfarlane-Dick, 2006, Clark, 2012).

3 Teaching context

The two programming courses were Programming Methods 3 (PM3, 2nd curricular year) & Programming Methods 4 (PM4, 3rd curricular year). Prior to these, students learned introductory programming in two previous courses, plus extra concepts in a Computational Logic course. These courses were provided in parallel (joint lectures, but separate hands-on lessons) to students in the IE and ICT programmes of studies. Course goals are described next.

In PM3: The goal is to introduce the students to large-scale programming concepts, one of the learning objective of the ACM/IEEE Computer Science Curricula (CSC). Specifically, students are introduced to the MVC architectural style, which divides programs among three blocks: the model (e.g., program state), the view (e.g., output), and the controller (e.g., program flow). The original MVC style proposal of Krasner & Pope (1988), which handles input in the controller, is contrasted with a more recent flavour proposed by Curry & Grace (2008), which handles input in the view (Nunes et al., 2015).

In PM4: The goal is for students to develop the knowledge and skills necessary to develop web applications. Students work with the client-server concept of web applications and study their operation, including analysis of the HTTP protocol and the processing of its messages by web clients and servers. PM4 includes data formats and metadata for web applications, including the meta-languages SGML and XML, and languages specified by them. It finishes with the study of various types of Web applications and the specific case of Web services.

4 The SimProgramming approach and learning assignment

The SimProgramming approach is based on four conceptual foundations: 1) business-like learning environment, 2) self-regulated learning; 3) co-regulation learning, and 4) formative assessment. Based on these, the learning activity process develops along four phases, and students have specific tasks in each phase (Pedrosa et al., 2016).

Learning assignment in PM3: For each team a specific problem using a software architecture is assigned, in order to stimulate and foster advanced programming skills. Students must develop a written document with a detailed explanation of the coding approaches they used to apply an MVC related architectural style involving different frameworks, libraries, and/or specific APIs (Morgado et al., 2012; Nunes et al., 2015).

Learning assignment in PM4: For each team a specific problem using protocols, web applications, and markup languages is assigned, to develop skills on the development of web applications. Students must develop a technological solution using a web system and a given web access platform (e.g. mobile devices), and explain in detail (including code examples) how it is possible to exchange information between systems using different markup languages.

Our development of the assignment in PM4, included some differences (detailed in Table 1) regarding the PM3 assignment described by Pedrosa et al. (2016):

Table 1 Differences between PM3 and PM4.

Variable	**PM3 (2012/ 2013)**	**PM4 (2013/2014)**
Professor	Professor A	Professor B
The impact of the assignment in the final grade	6/20	5/20
Students in the assignment	N=97 (IE:60; ICT:37)	N=49 (IE:32; ICT:17) – All, except 4 students, participated in PM3.
Established teams	N=15	N=9
Teams that concluded the assignment	N=13	N=9
Students with a final grade	N=66	N=49
Task changes	Weekly individual forms Reports (practitioners' on-line communities, status and final) Individual students forms for self and hetero-assessment	Fortnightly team forms There were no reports Meetings of tutors with teams for self and hetero-assessment

5 Methodology and data collection

We conducted semi-structured interviews (Cohen, Manion & Morrison, 2011) about self-regulation learning strategies applied by student in the assignment. The goal of the interviews was to have students describe their self-regulation learning strategies throughout the assignment. In the two research cycles 32 interviews were carried out.

The interview guide was constructed based on the literature (Zimmerman, 2008), and organized with five sets of questions/categories, including the one under scrutiny in this paper: *self-regulation learning strategies applied in the assignment.*

The interviews were conducted with students selected based on: role in the team (e.g. team leader), results in the assignment, special status (e.g. working-students). We also interviewed students whose assignment quality increased during the process.

Interviews were subjected to thematic analysis (Braun & Clarke, 2006). Content analysis matrixes were organized into categories, subcategories, indicators, and recording units (snippet sentences mentioned in interview. The content analysis matrixes were built based on the literature about SRL (Zimmerman, 2008, 2013), and during the analysis changes were made according to is identified in interviews, and we conducted a cyclical process of improvement, synthesis, and reflection.

The answers for the question/category set *"self-regulation learning strategies were applied in assignment"* were grouped into seven sub-categories about SRL strategies (Zimmerman, 2013), and are organized by indicators about the types of strategies that student adopted, identifying difficulties and factors that they believed have influenced their motivation. The indicators are the clear definitions/topic for each theme identified by answers in the interviews. After completing the content analysis matrixes, we organized and presented the results in tables, with the number of students who mentioned each indicator to identify what are the SRL strategies adopted by students.

6 Results and discussion

As shown on Table A.1, the most common strategy was information search, as in most interviews students said that they searched for content related to their work. Other strategies were mentioned less often. For example, few students explained what they did after retrieving information (information processing, information organization, and transformation or applicability of the information). We did find that some students said they had prepared a preliminary work plan. Another strategy mentioned was to follow guidelines provided by the teaching team. Only in Cycle 1 did some students (n=6) report not having any work planning strategy.

Table A.1 – Organizing, planning and transforming strategies.

Indicators	*Interviewed students*	
	Cycle 1 (N=21)	*Cycle 2 (N=11)*
A.1.1. Organizing - Information search	16	9
A.1.2. Organizing – Collected information	2	0
A.1.3. Planning – Work plan development	6	2
A.1.4. Planning - Following guidelines provided by tutors and teachers	3	2
A.1.5. Had no planned strategy	6	0
A.1.6. Transforming□- Drafting notes about collected information	2	0
A.1.7. Transforming□- Application of existing knowledge about the practice	1	0

Regarding time management strategies for the assignment, detailed in Table A.2, students mentioned several strategies, without any single one standing out. A few students explained that they worked during the week, and some mentioned devoting only one day per week. Procrastination was mainly mentioned in Cycle 1, decreasing in Cycle 2. Students explained that as being due to feeling the pressure (to deliver before the deadline) instilled on them to accomplish the task. They also explained being unable to manage their time due to other duties or responsibilities.

Table A.2 - Time management strategies.

Indicators	*Interviewed students*	
	Cycle 1 (N=21)	*Cycle 2 (N=11)*
A.2.1. Lack of time	3	2
A.2.2. Lack of time due to other responsibilities	1	0
A.2.3. Initiating the activity at the last moment (procrastination)	6	2

A.2.4. Organization on weekends	1	1
A.2.5. Submitted in next week	1	0
A.2.6. Realization of the assignment tasks at night	1	0
A.2.7. Grade impact factor of the assessment versus other courses	1	0
A.2.8. Division of tasks in the timeline	1	1
A.2.9. Devoted one day per week for resting	1	0
A.2.10. Placed extra activities aside	0	2
A.2.11. Working week to week	2	3
A.2.12. Devoted a specific day to work on the assignment	3	1

The difficulties students encountered while performing the assignment, as shown on Table A.3, were at the level of theoretical content and practical implementation of the assignment. Although the difficulties expressed by students were at the individual level, when asked about difficulties felt by their team they mentioned several. Some said that they hadn't experienced any difficulties carrying out the assignment. In Cycle 2, students specifically mentioned several difficulties.

Table A.3 - Identifying of the difficulties in the assignment.

Indicators	*Interviewed students*	
	Cycle 1 (N=21)	*Cycle 2 (N=11)*
A.3.1. Difficulties – Theoretical knowledge about the technology being studied	0	5
A.3.2. Difficulties – More than the previous year	0	1
A.3.3. Difficulties – The practical component implementation	0	1
A.3.4. Difficulties – Didn't find it complex to perform	5	4
A.3.5. Difficulties – Didn't experience any due to previous experience in PM3	0	1

The most mentioned strategy by students to resolve their difficulties, as shown on Table A.4, was the search for information (in Cycle 1, n=15; in Cycle 2, n=5), followed by seeking social assistance from both teacher (in Cycle 1, n=12; in Cycle 2, n=3) and peers (in Cycle 1, n=10; in Cycle 2, n=6). However, several Cycle 1 students (n=13) reported not having sought help: 2 students did not seek any help, 5 students did not seek help from the teacher and 5 students did not seek help from their peers. Some reasons pointed out for not seeking the teacher were feelings of shyness, shame, fear, or inferiority. In Cycle 2 students mentioned seeking social assistance and they do not mention the opposite (not seeking it, as it happened in Cycle 1). There were students who gave up trying to solve the difficulties, e.g. did not clarify their doubts/problems. Working-students explained their difficulties seeking help from colleagues as being due to differences in work patterns.

Table A.4 - Resolution of difficulties strategies.

Indicators	*Interviewed students*	
	Cycle 1 (N=21)	*Cycle 2 (N=11)*
A.4.1. Seeking Social Assistance (SOA)– Did not seek any	2	0
A.4.2. SOA – Did not seek help from teacher	5	2
A.4.3. SOA - Did not seek help from peers	5	1
A.4.4. SOA - Did not seek help due to different work pace (worker-student)	1	0
A.4.5. Resolution of Difficulties (RD)– Information search	15	5
A.4.6. RD - Use of practical exercises	1	1
A.4.7. SOA – Teachers	12	3
A.4.8. SOA – Team peers	10	6
A.4.9. SOA - Senior colleagues	3	1
A.4.10. SOA – Family member	1	0
A.4.11. SOA – Others	1	1
A.4.12. RD - Gave up seeking help, even after finding information from searching	2	2

On the review of the tasks required for the assignment, as shown on Table A.5, students' most common strategies were checking for misspellings, revising sentence construction, and correcting theoretical content. In Cycle 2, the focus of the review was more on content, instead of the mostly superficial revisions of Cycle 1. Some students from Cycle 1 stated not having reviewed their work, something which in Cycle 2 was not mentioned. A fact that emerged in Cycle 2 was a student reporting to have done the review by reflecting upon comments that were given by teacher to other teams.

Table A.5 - Work review strategies.

Indicators	*Interviewed students*	
	Cycle 1 (N=21)	*Cycle 2 (N=11)*
A.5.1. No revision made	5	0
A.5.2. Component programming practice	1	2
A.5.3. Additional information	2	1
A.5.4. Typo correction and sentence construction fixes	8	2
A.5.5. Overall review and general changes	9	1
A.5.6. Portuguese-language errors and content corrections	6	4
A.5.7. Global content (not specific)	1	0
A.5.8. Avoiding repetition of information	1	0
A.5.9. Reflecting feedback provided by teacher to other teams	0	1

As shown on table A.6, students expressed several factors has having affected their motivation, either positively or negatively (lack of motivation). Most are of a personal nature, but some are linked to interpersonal and social dimensions. For example, the completion of the programme of studies, and the perception that the work contributed to learning and skill development. The grade impact of the assignment was the single aspect most mentioned by students has having had an influence on motivation.

In Cycle 2, students mentioned other factors has having influenced their motivation, including believing that the assignment helped develop interpersonal skills, programming skills, and also that it helped understand the content of course (n=3). In addition, students in Cycle 2 almost did not mention factors has having created lack of motivation: only one student mentioned lack of motivation due to having a grade (in others grades - tests) too low to complete the course.

Table A.6 - Factors influencing the motivation.

Indicators	*Interviewed students*	
	Cycle 1 (N=21)	*Cycle 2 (N=11)*
A.6.1. Finish the course programme of studies	6	2
A.6.2. Completing the PM3 or PM4 course	4	1
A.6.3. Maintaining their scholarship fund	1	0
A.6.4. Showing the ability to complete the activity	1	0
A.6.5. Achieving good results	3	1
A.6.6. Interest in programming	2	1
A.6.7. Comply with an obligation	1	2
A.6.8. Grade impact of the assignment	17	8
A.6.9. To present work that he/she agreed to do	1	0
A.6.10. Throwback to previous life experience	1	0
A.6.11. Responsibility for teamwork	1	0
A.6.12. Found the process interesting (SimProgramming approach)	1	0
A.6.13. Learning	3	3
A.6.14. Preparation for the labor market	2	2
A.6.15. Avoiding the stress of procrastination	1	0
A.6.16. Working in new team	0	1

A.6.17. Assignment helps to develop programming skills	0	9
A.6.18. Assignment helps develop interpersonal skills	0	4
A.6.19. Assignment helps to PM3 or PM4 course	0	3
A.6.21. Lack of motivation – Is tired (of studying)	1	0
A.6.22. Lack of motivation – Overall grades are not enough to complete the course	1	1
A.6.23. Lack of motivation – Didn't learned anything new or useful	5	0
A.6.24. Lack of motivation - Wanted to learn but could not	1	0

In both cycles, most students working alone said they preferred to do it at home (see table A.7). Other places were mentioned, like the library or city bars. As for concentration strategies, there is a preference for quiet places. However, no explanation was given by the students on specific strategies they used to stay focused.

Table A.7 - Environmental structuring strategies.

Indicators	*Interviewed students*	
	Cycle 1 (N=21)	*Cycle 2 (N=11)*
A.7.1. The Physical Setting (TPS)– Home	16	7
A.7.2. TPS – Library	4	2
A.7.3. TPS - Multiple locations	3	2
A.7.4. Concentration Strategy (CS) - Silence	6	1
A.7.5. CS - Studying with music	3	1
A.7.6. CS - Background noise	2	0
A.7.7. CS – Place without distractions	5	2
A.7.8. CS – Inconstant	2	1

7 Conclusions and future work

In this work, students mentioned applying several self-regulated learning strategies, such as: information search; work reviewing; time management; social seeking assistance; resolution of difficulties; and environmental structuring. However, other strategies were mentioned by fewer students, namely: strategies about information processing, organization, or application; self-reflection and self-assessment; psychological strategies to improve attention; and awareness of their difficulties.

From Cycle 1 to Cycle 2, we verified improvements in strategies for time management, work review, social assistance seeking, difficulty awareness, and motivational factors related to skill development. We hypothesize that participation in PM3 activities using the SimProgramming approach may have helped students develop the self-regulation learning strategies that emerged in Cycle 2.

For example, the work reviewing strategies improved from Cycle 1 to Cycle 2: students mentioned more specific content revisions. Also, in Cycle 2, students demonstrated some awareness of their difficulties, something not seen in Cycle 1. In Cycle 1 there were students who did not seek assistance, reportedly due to factors such as shyness, fear, or shame. No such factors were mentioned in Cycle 2. Procrastination behaviours were also recorded in Cycle 1, but in Cycle 2 procrastination was no longer mentioned by the students.

The motivational factors also influenced the students' perception about their self-regulation learning and progress in the assignment. In Cycle 2, the students felt that the assignment helped develop programming and interpersonal skills, and contributed to a better understanding of the content of course.

We suggest the development of activities to raise awareness in students about the various types of strategies that can be pursued for success in academic tasks. Further research about the SimProgramming approach should strive to achieve a better understanding of the impact of team work and assignment grade on the self-regulation of student learning.

8 Acknowledgments

Pedrosa, D. thanks the Fundação para a Ciência e Tecnologia (FCT), Portugal, for Ph.D. Grant SFRH/BD/87815/2012. We would like to thank all the students and teachers who collaborated on this research.

9 References

Alharbi, A., Paul, D., Henskens, F., & Hannaford, M.(2011). An investigation into the learning styles and self-regulated learning strategies for computer science students. In Proceedings ascilite.

Association for Computing Machinery (ACM) & IEEE Computer Society (IEEE-CS), 2008. "Computer Science Curriculum 2008: An Interim Revision of CS 2001." Computing Curriculum Series [on-line],http://www.acm.org//education/curricula/ComputerScience2008.pdf.

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

Braun, V., & Clarke, V. (2006). Using thematic analysis in psychology. *Qualitative research in psychology*, 3(2), 77-101.

Cazan, A. M. (2013). Teaching self regulated learning strategies for psychology students. *Procedia-Social and Behavioral Sciences*, 78, 743-747.

Cagiltay, N. E. (2007). Teaching software engineering by means of computer-game development: Challenges and opportunities. *British Journal of Educational Technology*, *38*(3), 405-415.

Clark, I. (2012). Formative Assessment: Assessment Is for Self-regulated Learning. *Educational Psychology Review*, 24(2), 205–249.

Cohen, L; Manion, L; Morrison, K. (2011). *Research Methods in Education*. 7th Edition. London, Routledge-Taylor & Francis Group.

Curry, E., & Grace, P. (2008). Flexible self-management using the model-view-controller pattern. *Software*, IEEE, 25(3),84-90.

Fernández, E., Bernardo, A., Suárez, N., Cerezo, R., Núñez, J. C., & Rosário, P. (2013). Predicción del uso de estrategias de autorregulación en educación superior. *Anales de psicología*, 29(3), 865-875.

Gomes, A., & Mendes, A. J. (2007). Learning to program-difficulties and solutions. In *International Conference on Engineering Education–ICEE*, 2007, Coimbra, Portugal.

Hattie, J., & Timperley, H. (2007). The Power of Feedback. *Review of Educational Research*, 77(1), 81–112.

Lahtinen, E., Ala-Mutka, K., & Järvinen, H. M. (2005). A study of the difficulties of novice programmers. *In ACM SIGCSE Bulletin*, 37(3), 14-18.

Liu, Y., & Phelps, G. (2011). Challenges and professional tools used when teaching web programming. *Journal of Computing Sciences in Colleges*, *26*(5), 116-121.

Morgado, L.; Fonseca, B.; Martins, P.; Paredes, H.; Cruz, G.; Maia, A. M.; Nunes, R.; Santos, A. (2012). Social networks, microblogging, virtual worlds, and Web 2.0 in the teaching of programing techniques for software engineering: a trial combining collaboration and social interaction beyond college. In *Global Engineering Education Conference (EDUCON)*.1-7. IEEE. doi:10.1109/EDUCON.2012.6201129.

Nicol, D. J., & Macfarlane-Dick, D. (2006). Formative assessment and self-regulated learning: a model and seven principles of good feedback practice. *Studies in Higher Education*, 31(2), 199–218.

Nunes, R. R., Pedrosa, D., Fonseca, B., Paredes, H., Cravino, J., Morgado, L., & Martins, P. (2015). Enhancing students' motivation to learn software engineering programming techniques: a collaborative and social interaction approach. In *Universal Access in Human-Computer Interaction*. *Access to Learning, Health and Well-Being*. 189-201. Springer International Publishing.

Pedrosa, D.; Cravino, J.; Morgado, L.; Barreira, C.; Nunes, R.R.; Martins, P. & Paredes, H. (2016). Simprogramming: the development of an integrated teaching approach for computer programming in higher education. In *Proceedings 10th annual International Technology, Education and Development Conference (INTED 2016)*. Valencia, Spain.

Pintrich, P. R. (2004). A conceptual framework for assessing motivation and self-regulated learning in college students. *Educational psychology review*,16(4), 385-407.

Robins, A., Rountree, J. & Rountree. N. (2003). Learning and teaching programming: A review and discussion. *Computer Science Education*, 13(2), 137–172.

Zimmerman, B. J., & Schunk, D. H. (2007). Motivation: An essential dimension of selfregulated learning. In D. H. Schunk & B. J. Zimmerman (Eds.), *Motivation and self-regulated learning: Theory, research, and applications* (pp. 1–30). Mahwah, NJ: Lawrence Erlbaum.

Zimmerman, B. J. (2008). Investigating Self-Regulation and Motivation: Historical Background, Methodological Developments, and Future Prospects. *American Educational Research Journal*, 45(1), 166–183.

Zimmerman, B. J. (2013). From cognitive modeling to self-regulation: A social cognitive career path. Educational Psychologist, 48(3), 135-147.