

Contents

1	Introduction	1
1.1	Teaching Software Engineering with a Practical Approach	2
1.1.1	Types of SE Project Courses	3
1.1.2	Challenges for Implementing Practical Experiences in SE Courses	4
1.2	Problem Statement	5
1.3	Work Hypotheses	6
1.4	Objectives	6
1.5	Research Methodology	8
1.5.1	Information Gathering	8
1.5.2	Definition of the EduProcess	9
1.5.3	Evaluation of the EduProcess	9
1.6	Summary of the Thesis Document	10
2	Related Work	11
2.1	Teaching Software Engineering	12
2.1.1	Expositive Methods	12
2.1.2	Inverted Classroom	13
2.1.3	Problem-Based Learning Method	13
2.1.4	Hands-on Methods (Learn-by-Doing)	13
2.1.4.1	Hands-on Labs	14
2.1.4.2	Hands-on Simulated Labs	15
2.1.5	Hands-on Remote Labs	16
2.1.6	Service-Learning	16
2.1.7	Case Studies Method	17
2.2	Definition of Software Processes for Academic Scenarios	17
2.2.1	Disciplined Software Processes	18
2.2.2	Agile Software Development Processes	23
2.3	Matching Instructional Approaches And Software Processes	26
2.3.1	Instructional Approaches Considered in the Systematic Study	26
2.3.2	Software Processes Considered in the Systematic Study	27
2.3.3	Mapping Instructional Approaches and Software Processes	27
2.4	Instructional Practices for Software Engineering Courses	29

3	Identifying the SE Best Practices to Be Taught in Undergraduate Programs	33
3.1	ACM-IEEE SE Curriculum Guidelines	33
3.2	SE Best Practices	35
3.2.1	SE Best Practices Being Taught	36
3.2.2	Methodology	36
3.2.3	Results	38
3.3	Software Engineering Education in Chile	43
3.3.1	SE Course Characterization	44
3.3.2	Computer Science Courses	45
3.3.3	SE Knowledge Areas Coverage	45
3.3.4	Courses Evaluations	47
3.3.5	SE Course Temporality	50
3.3.6	Analysis of SE Teaching in Chile	50
3.4	Practices for SE Education	52
4	EduProcess	55
4.1	General Approach	55
4.1.1	Target Software Product	56
4.1.2	Supported Roles	56
4.2	Structure of EduProcess	57
4.2.1	A Brief Description of the Software Process	59
4.2.1.1	Conception	59
4.2.1.2	Iterations 1 and 2	60
4.2.1.3	Deployment	61
4.2.1.4	Work Dynamics	62
4.2.2	The Instructional Process	62
4.2.2.1	Preparedness	63
4.2.2.2	Conception	64
4.2.2.3	Iteration 1 and 2	65
4.2.2.4	Deployment	66
4.2.2.5	Reflexive Weekly Monitoring	66
4.2.2.6	Weekly Work Meetings with Clients and Users	67
4.3	SE Best Practices Included in EduProcess	68
4.4	Process Restrictions	68
4.4.1	Type of Projects to be Addressed	69
4.4.2	Product Type, Size and Complexity	70
4.4.3	Dealing with the Requirements for Academic Software Processes	70
5	EduProcess Evaluation	72
5.1	Case Study Scenario	73
5.2	Data Gathering and Analysis	74
5.2.1	Quantitative Data	74
5.2.1.1	Course Grades	74
5.2.1.2	Project Grades	75
5.2.1.3	Student Survey	75

5.2.1.4	Peer-Assessment	75
5.2.1.5	Amount of Software Built	75
5.2.1.6	Suitability of the Project Implementation	76
5.2.2	Qualitative Data	76
5.2.2.1	Peer-Assessments	76
5.2.2.2	Monitoring Feedback	76
5.3	Results	77
5.3.1	Course and Project Grades	77
5.3.2	End of Semester Student Surveys	80
5.3.3	Peer-Assessments	82
5.3.3.1	Coordination Level	82
5.3.3.2	Sense of Team Belonging	84
5.3.3.3	Team Members Engagement	84
5.3.3.4	Initiative	85
5.3.3.5	Communication	85
5.3.3.6	Commitment	85
5.3.4	Evaluation of Team Productivity and Effectiveness	87
5.4	Discussion	90
5.5	Threats to Validity	93
5.5.1	Construct Validity	93
5.5.2	Reliability	94
5.5.3	Internal Validity	94
5.5.4	External Validity	94
6	Conclusions and Future Work	95
6.1	Summary of the Thesis Work	95
6.2	Conclusions	96
6.3	Contributions	97
6.4	Future Work	97
	Bibliography	98
	Appendix A Software Process Definition	113
A.1	Project Contexts	113
A.1.1	Type of Projects	113
A.1.2	Product Type, Size and Complexity	114
A.1.3	Our Context	115
A.2	Roles	117
A.2.1	Project Manager	117
A.2.2	Analyst	118
A.2.3	Designer/Developer	118
A.2.4	Quality Assurance Engineer	118
A.3	Introduction to the Phases, Activities and Milestones	119
A.3.1	Conception	120
A.3.2	Iteration 1	120
A.3.3	Iteration 2	120
A.3.4	Deployment	120

A.4	Conception Phase	121
A.4.1	Introduction	121
A.4.2	Inputs	121
A.4.3	Activities	121
A.4.3.1	Problem Exploration	121
A.4.3.2	Main User Requirements Identification	123
A.4.4	Reviews (Milestones)	123
A.4.4.1	Problem Presentation	123
A.4.4.2	Prototype Presentation	124
A.4.5	Outputs	124
A.4.6	Deliverables	124
A.5	Iteration 1 and 2	124
A.5.1	Introduction	124
A.5.2	Inputs	125
A.5.3	Activities	125
A.5.3.1	Project Management	127
A.5.3.2	Prototype Evolution	127
A.5.3.3	User Requirements Definition	128
A.5.3.4	Software Requirements Definition	131
A.5.3.5	Product Design	134
A.5.3.6	Product Implementation	135
A.5.3.7	Test Case Specification	136
A.5.3.8	System Verification	138
A.5.3.9	User Validation	138
A.5.3.10	Deployment	138
A.5.4	Reviews	139
A.5.4.1	Requirements Evaluation	139
A.5.4.2	Design Evaluation	140
A.5.4.3	Demo Presentation	140
A.5.5	Outputs	140
A.5.5.1	Project Plan	140
A.5.5.2	User Requirements	140
A.5.5.3	Software Requirements	140
A.5.5.4	Test Cases	141
A.5.5.5	Requirements Review Notes	141
A.5.5.6	Design Review Notes	141
A.5.5.7	Demo Review Notes	141
A.5.6	Deliverables	141
A.5.6.1	Software Requirement Document	141
A.5.6.2	Design Document	143
A.5.6.3	Software Product - SPr	143
A.6	Software Deployment	143
A.6.1	Introduction	143
A.6.2	Inputs	144
A.6.3	Activities	144
A.6.3.1	Tuning of the Software Product	144
A.6.3.2	Product Deployment	144

A.6.3.3	Client/User Assisted Usage	145
A.6.3.4	Incidence Registration	145
A.6.4	Reviews	145
A.6.5	Outputs and Deliverables	145
Appendix B	Instructor Software Process Definition	146
B.1	Phases, Activities and Milestones	146
B.2	Preparedness	146
B.3	Conception	149
B.3.1	Inputs	149
B.3.2	Activities	149
B.3.2.1	Week 1	149
B.3.2.2	Remaining Weeks	149
B.3.3	Reviews	151
B.3.3.1	Problem Presentation	151
B.3.3.2	Prototype Presentation	151
B.3.3.3	Peer Evaluation 1	151
B.3.4	Outputs	152
B.3.5	Deliverables	152
B.4	Iteration 1	152
B.4.1	Inputs	152
B.4.2	Activities	152
B.4.3	Reviews	154
B.4.3.1	Requirements Evaluation	154
B.4.3.2	Design Evaluation	154
B.4.3.3	Demo Presentation	155
B.4.3.4	Peer Evaluation 2	155
B.4.4	Outputs	155
B.4.5	Deliverables	155
B.5	Iteration 2	155
B.5.1	Inputs	155
B.5.2	Activities	155
B.5.3	Reviews	157
B.5.3.1	Requirements Evaluation	157
B.5.3.2	Design Evaluation	157
B.5.3.3	Demo Presentation	157
B.5.3.4	Peer Evaluation 3	157
B.5.4	Outputs	157
B.5.5	Deliverables	157
B.6	Deployment	158
B.6.1	Inputs	158
B.6.2	Activities	158
B.6.3	Reviews	158
B.6.4	Outputs	158
B.6.5	Deliverables	158
Appendix C	Reflexive Weekly Monitoring	159

C.1	RWM - Structure	160
C.2	Monitoring Schedule	162
C.2.1	Conception	162
C.2.2	Iteration 1	164
C.2.3	Iteration 2	165
Appendix D SSP versus EduProcess		167
D.1	SSP	167
D.1.1	Phases	167
D.1.2	Roles and Context	168
D.2	Comparison	168
Appendix E A Systematic Mapping Study on Practical Approaches to Teaching Software Engineering - FIE 2014		171

List of Tables

1.1	Comparison of the Industrial and Academic Development Scenarios	7
2.1	Disciplined Software Processes for the Academia	19
2.2	Comparison of Disciplined Processes with the Set of “Minimal” Requirements for Academic Process	21
2.3	Agile Software Processes for the Academia	23
2.4	Comparison of Agile Processes with the Set of “Minimal” Requirements for Academic Process	24
3.1	Specific Areas Considered in the 2004 - ACM-IEEE SE Curriculum Guidelines	34
3.2	Specific Areas Considered in the 2014 - ACM-IEEE SE Curriculum Guidelines	35
3.3	Search Strings	37
3.4	Specific Search Strings for Each Digital Library	37
3.5	Number of the Papers Retrieved from Each Digital Library	38
3.6	Results After the Use of the Guidelines	39
3.7	Primary Studies Selected	39
3.8	Reported Best Practices (to be applied by students)	40
3.9	Reported Best Practices (instructional)	40
3.10	Best Practices Reported in More than One PS	41
3.11	How was Validated the Use of the Best Practices (TiP - Taught in Practice) .	41
3.12	ACM-IEEE SE2014 Curriculum x Reported “Best Practices”	42
3.13	Number of Computer-Related Programs Offered by Chilean Universities in 2015	44
3.14	Number of Software Engineering Courses in Each Program in 2015	46
3.15	Most Frequent Knowledge Units in CS Engineering Programs	47
3.16	Least Frequent Knowledge Units in CS Engineering Programs	47
3.17	Most Frequent Knowledge Units in CS Technology Programs	48
3.18	Least Frequent Knowledge Units in CS Technology Programs	48
3.19	Mandatory Practices	52
3.20	Recommended Practices	53
3.21	Optional Practices	53
4.1	SE Best Practices Included in the EduProcess	69
5.1	Semesters and Students Enrolled in the CC5401 Course	78
5.2	Statistical Analysis of Course and Project Grades	78
5.3	Items of the Student Survey	80
5.4	Student Survey Results	81

5.5	Statistical Analysis of the Peer-Assessments	82
5.6	Team Aspects - Coordination, Sense of Team Belonging, Engagement	83
5.7	Team Aspects - Initiative, Communication, Commitment	86
5.8	Team Productivity	88
5.9	Team Members Productivity and Effectiveness	89
5.10	Statistical Analysis of FP	90
5.11	Team Productivity Detail	91
5.12	Project Deployment (meaning: ✗ - not deployed, ✓ - deployed)	92
A.1	Context Restrictions	116
A.2	Software Life Cycle Model	119
A.3	User Requirements Specification Detail	129
A.4	Software Requirements Specification Detail	133
A.5	Test Case Detail	137
A.6	Review Notes Detail	141
B.1	Software Life Cycle Model	147
D.1	Comparison of SSP and EduProcess with the Set of “Minimal” Requirements for an Academic Software Process	169

List of Figures

1.1	General Scenario of Software Development Experiences in SE Courses	3
1.2	Summary of the Research Methodology	8
2.1	Systematic Mapping Process by Petersen et al. [126]	26
2.2	SE Instructional Approaches Reported by Year	28
2.3	Instructional Approaches Reported	30
3.1	Course Characterization in (a) CS Engineering Programs and (b) CS Technology Programs	46
3.2	Knowledge Areas Characterization of CS Engineering Programs	48
3.3	Knowledge Areas Characterization of CS Technology Programs	49
3.4	Courses Evaluation of CS Engineering Programs	49
3.5	Courses Evaluation of CS Technology Programs	49
3.6	Course Temporality in CS Engineering Programs	50
3.7	Course Temporality in CS Technology Programs	51
4.1	General Architecture of EduProcess	55
4.2	EduProcess Tracks	56
4.3	Architecture Assumed for an Information System	56
4.4	Structure of EduProcess	58
4.5	Structure of the Software Process	59
4.6	Structure of the Software Prototype	60
4.7	Relationship Among the Work Products of Iterations 1 and 2.	60
4.8	Structure of the Instructional Process	63
4.9	General Workflow of Iterations 1 and 2 for the Instructional Process	66
4.10	General Structure of the RWM Sessions	67
4.11	Process of Co-work between the Software Team and the Stakeholders	68
5.1	Timeline of the Evaluation Process	73
5.2	Course Grades using (a) SSP and (b) EduProcess	79
5.3	Project Grades using (a) SSP and (b) EduProcess	79
A.1	General Structure of EduProcess	120
A.2	Conception Phase	121
A.3	Problem Exploration	122
A.4	Main User Requirements	123
A.5	Phase - Iteration 1 and 2	126

A.6	Project Management Activity	127
A.7	Prototype Evolution Activity	128
A.8	User Requirement Activity	130
A.9	Software Requirement Activity	132
A.10	Design Activity	135
A.11	Implementation Activity	136
A.12	Test Case Activity	137
A.13	System Verification Activity	137
A.14	User Validation Activity	138
A.15	Optional Deployment Iteration 1	139
A.16	Deployment	144
B.1	EduProcess-Instructional	146
B.2	EduProcess-Instructional- Preparedness Phase	148
B.3	EduProcess-Instructional - Conception Phase	150
B.4	EduProcess-Instructional- Iteration 1	153
B.5	EduProcess-Instructional- Iteration 2	156
B.6	EduProcess-Instructional- Deployment	158
C.1	Structure of a RWM Session.	161
D.1	Structure of SSP	167