TABLE OF CONTENT

DECLARATION BY THE SCHOLAR iv
SUPERVISOR’S CERTIFICATE vi
ACKNOWLEDGEMENTS xiv
ABSTRACT xvi
LIST OF FIGURES xviii
LIST OF TABLES xix

CHAPTER-1

INTRODUCTION 1
1.1 BASIS FOR THE NEED FOR REFORMS IN COMPUTING 5
EDUCATION
1.2 EVOLUTION OF SOFTWARE DEVELOPMENT EDUCATION 9
1.3 RESEARCH APPROACH 28
1.4 THESIS LAYOUT 33

CHAPTER-2

IDENTIFICATION OF CORE COMPETENCIES FOR SOFTWARE ENGINEERS 35
2.1 STUDY REPORT ON CORE COMPETENCIES FOR ENGINEERS WITH SPECIFIC REFERENCE TO SOFTWARE ENGINEERING 35
2.2 NECESSARY COMPETENCIES AS EDUCATIONAL OUTCOMES FOR SOFTWARE ENGINEERS AS RECOMMENDED BY ACCREDITATION BOARDS, PROFESSIONAL SOCIETIES’ AND OTHER APPROACHES 39
2.2.1 IMPACT ON CURRICULUM AND FUTURE DIRECTIONS 40
2.2.2 INDIAN SCENARIO 41
2.3 SOME OTHER CONTEMPORARY RECOMMENDATIONS ABOUT DESIRED COMPETENCIES OF ENGINEERING GRADUATES 42
2.4 RECOMMENDATIONS OF SOME INTERNATIONAL PROFESSIONAL SOCIETIES RELATED TO COMPUTING 44
2.5 SOME CONTEMPORARY RECOMMENDATIONS ON DESIRED COMPETENCIES OF SOFTWARE DEVELOPERS 47
2.6 A PERSPECTIVE FROM THE PROFESSIONAL CODES OF CONDUCT, ETHICS, AND/OR PRACTICE 51
2.7 CLASSICAL AND CONTEMPORARY RECOMMENDATIONS ON DESIRED COMPETENCIES OF GRADUATES 53
2.8 A COMPREHENSIVE DISTILLED VIEW ON DESIRED COMPETENCIES 56
2.9 FURTHER EMPIRICAL INVESTIGATIONS ON REQUIRED CORE COMPETENCIES FOR ENGINEERING GRADUATES WITH REFERENCE TO THE INDIAN IT INDUSTRY 56
2.10 CLASSIFYING THE CORE COMPETENCIES FOR SOFTWARE DEVELOPERS 58
2.11 CHAPTER CONCLUSION 61

CHAPTER-3

DISTINGUISHING FEATURES OF SOFTWARE DEVELOPMENT AND REQUISITE TAXONOMY OF CORE COMPETENCIES 64

3.1 PROGRAMMING AS AN ART TO SOFTWARE ENGINEERING 65
3.2 DEBUGGING AS A CORE ACTIVITY IN SOFTWARE DEVELOPMENT 67
3.3 PROCESS CENTRIC SYSTEM DEVELOPMENT AND MAINTENANCE IN SOFTWARE ENGINEERING 68
3.4 SOFTWARE AS INTEGRAL PART OF BUSINESS, AND NEED FOR COMPREHENSION FOR SOFTWARE MAINTENANCE 68
3.5 ROLE OF EMPATHY AND SOCIAL SENSITIVITY IN SOFTWARE DEVELOPMENT 69
3.6 PROJECT SCOPING AND ESTIMATION FOR SOFTWARE CONTRACT 71
3.7 LEARNING NEW DOMAIN AND KNOWLEDGE STRUCTURING IN SOFTWARE DEVELOPMENT 71
3.8 SOFTWARE DEVELOPMENT PROCESS FOR ILL-DEFINED PROBLEMS 72
3.9 EMPIRICAL AND QUALITATIVE APPROACHES IN SOFTWARE DEVELOPMENT RESEARCH 74
3.10 SOFTWARE DEVELOPMENT: WHOLE-BRAIN ACTIVITY 75
3.11 REVISED TAXONOMY OF CORE COMPETENCIES FOR SOFTWARE DEVELOPERS 76

CHAPTER-4

SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF BASIC COMPETENCIES 82

4.1 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF TECHNICAL COMPETENCE 83
4.2 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF COMPUTATIONAL THINKING 91
4.3 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF DOMAIN COMPETENCE 98
4.4 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF COMMUNICATION COMPETENCE 106
4.5 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF COMPLEX PROBLEM SOLVING COMPETENCE
   4.5.1 EXPERT PROBLEM SOLVERS 118
4.6 CHAPTER CONCLUSION 123

CHAPTER-5

SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF COMPETENCY DRIVER-HABITS OF MIND 125
5.1: SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF ATTENTION TO DETAILS 126
5.2: SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF CRITICAL AND REFLECTIVE THINKING 130
5.3: SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF CREATIVITY AND INNOVATION 138
5.4: CHAPTER CONCLUSION 144

CHAPTER-6

SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF COMPETENCY CONDITIONING ATTITUDES AND PERSPECTIVES 145
6.1 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF CURIOSITY 146
6.2 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF DECISION MAKING PERSPECTIVE 154
6.3 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF SYSTEMS-LEVEL PERSPECTIVE 165
6.4 SOFTWARE DEVELOPERS’ EDUCATION FOR DEVELOPMENT OF INTRINSIC MOTIVATION TO CREATE/IMPROVE ARTIFACTS 175
6.5 CHAPTER CONCLUSION 181
CHAPTER-7

THE PHENOMENON OF ‘LEARNING’ 182

7.1  EMPIRICAL INVESTIGATIONS FOR ASSESSING EFFECTIVENESS OF EDUCATIONAL METHODS WITH RESPECT TO THE REQUIREMENTS OF SOFTWARE DEVELOPMENT

7.1.1  EMPIRICAL STUDIES ON EFFECTIVENESS OF TEACHING METHODS AND EDUCATIONAL EXPERIENCES OF COMPUTING STUDENTS AND SOFTWARE DEVELOPERS 182

7.1.2  EMPIRICAL EXAMINATION OF SOFTWARE DEVELOPMENT EDUCATION THROUGH BLOOM’S TAXONOMY 187

7.1.3  QUALITATIVE STUDY OF EFFECTIVE LECTURES 191

7.1.3.1  PERCEPTIONS OF COMPUTING STUDENTS AT SENIOR AND JUNIOR LEVELS 191

7.1.3.2  PERCEPTIONS OF FACULTY MEMBERS IN ENGINEERING INSTITUTES 192

7.1.4  QUANTITATIVE STUDY OF EFFECTIVE LECTURES 193

7.1.4.1  PERSPECTIVE OF COMPUTING STUDENTS 194

7.2  REFLECTIONS ABOUT THE PHENOMENON OF ‘LEARNING’ 197

7.3  IMPLICATIONS FOR SOFTWARE DEVELOPMENT EDUCATION 199

7.4  STUDENT ENGAGEMENTS FOR FACILITATING DEEP LEARNING THROUGH HIGHER EDUCATION

7.4.1  CURRICULUM INTEGRATION 202

7.4.2  SOLO TAXONOMY 205

7.4.3  COLLABORATIVE LEARNING 206

7.4.3.1  PAIR PROGRAMMING 209

7.4.4  CROSS-LEVEL PEER MENTORING 211

7.4.4.1  POSSIBILITY OF CROSS-LEVEL PEER MENTORING IN SOFTWARE DEVELOPMENT EDUCATION 214

7.5  CHAPTER SUMMARY 215

CHAPTER-8

A FRAMEWORK OF PEDAGOGIC ENGAGEMENTS IN SOFTWARE DEVELOPMENT EDUCATION 216

8.1  THREE-DIMENSIONAL KNOWLEDGE DOMAIN FOR DESIGNING COMPUTING COURSES 218

8.2  TWO CORE PRINCIPLES RELATED TO LEARNING 221

8.2.1  COGNITIVE DISSONANCE 221
8.2.2 COGNITIVE FLEXIBILITY

8.3 FOUR-DIMENSIONAL TAXONOMY OF PEDAGOGIC ENGAGEMENTS IN SOFTWARE DEVELOPMENT EDUCATION

8.3.1 DIMENSION 1- LEVELS OF ACTIVE ENGAGEMENTS (EXTENSION OF BLOOM’S TAXONOMY)

8.3.2 DIMENSION 2- LEVELS OF INTEGRATIVE ENGAGEMENTS (EXTENSION OF SOLO TAXONOMY)

8.3.3 DIMENSION 3- LEVELS OF REFLECTIVE ENGAGEMENTS

8.3.4 DIMENSION 4- LEVELS OF COLLABORATIVE ENGAGEMENTS

8.4 CHAPTER SUMMARY

CHAPTER-9

SOME INTERVENTIONS FOR ENHANCING THE QUALITY OF SOFTWARE DEVELOPMENT EDUCATION

9.1 INCREASING COGNITIVE DISSONANCE THROUGH A PROBLEM-CENTRIC APPROACH IN SOFTWARE DEVELOPMENT EDUCATION

9.1.1 INQUIRY TEACHING IN SOFTWARE DEVELOPMENT EDUCATION

9.1.1.1 SERO MODEL FOR INQUIRY TEACHING IN SOFTWARE DEVELOPMENT EDUCATION

9.1.2 PROJECT-INCLUSIVE TEACHING IN SOFTWARE DEVELOPMENT EDUCATION

9.1.3 CREATING CONDITIONS FOR REFLECTIVE ENGAGEMENTS IN SOFTWARE DEVELOPMENT EDUCATION

9.2 INCREASING COGNITIVE FLEXIBILITY THROUGH A MULTIFACETED INTEGRATED APPROACH IN SOFTWARE DEVELOPMENT EDUCATION

9.2.1 MULTILEVEL INFUSION FOR CONTINUOUS INTEGRATION IN SOFTWARE DEVELOPMENT EDUCATION

9.2.2 INTEGRATIVE CAPSTONE COURSES IN SOFTWARE DEVELOPMENT EDUCATION

9.2.3 GROUP AND COMMUNITY ORIENTED ENGAGEMENTS IN SOFTWARE DEVELOPMENT EDUCATION

9.2.3.1 COLLABORATIVE PAIR AND QUADRUPLE PROGRAMMING

9.2.3.2 CROSS-LEVEL PEER MENTORING IN SOFTWARE DEVELOPMENT EDUCATION

9.3 REFLECTIVE WORKSHOP ON PEDAGOGY FOR ENGINEERING FACULTY
CHAPTER-10

SUMMARY AND FUTURE SCOPE OF WORK

REFERENCES

APPENDICES
A1 SPINE-LIKE SURVEY ON IMPORTANCE OF COMPETENCIES
A2 A COMPREHENSIVE DISTILLED VIEW ON DESIRED COMPETENCIES
A3 REVISED SURVEY ON REQUIRED COMPETENCIES, 2007
A4 MAPPING OF THIRTY-FIVE COMPETENCIES (APPENDIX A3) WITH FINAL SET OF TWELVE CORE COMPETENCIES
A5 CATALOGUE OF TECHNICAL AND TECHNICALLY ORIENTED ACTIVITIES RELATED TO SOFTWARE DEVELOPMENT
A6 TAXONOMY OF COMMON SOFTWARE BUGS
A7 PROPOSED CURRICULUM FOR MASTERS IN ARCHAEO-HERITAGE INFORMATICS
A8 SOME SUGGESTIONS FOR BREADTH COURSES
A9 INADEQUATE DEVELOPMENT OF CURIOSITY IN SOFTWARE DEVELOPMENT EDUCATION
A10 SURVEY: “SOFTWARE DEVELOPERS - (HOW) DID YOUR COLLEGE HELP YOU IN YOUR DEVELOPMENT?”
A1 EFFECTIVENESS OF TEACHING METHODS: SURVEY OF SOFTWARE DEVELOPERS (2009)
A11 EMPIRICAL EXAMINATION OF SOFTWARE DEVELOPMENT EDUCATION THROUGH BLOOM’S TAXONOMY
A12 ANECDOTES OF MOST EFFECTIVE LEARNING EXPERIENCES/LECTURES
A13 QUANTITATIVE STUDY OF COMPUTING STUDENTS’ PERSPECTIVE OF EFFECTIVE LECTURES
A14 SUMMARY OF SERO STYLE LECTURES IN TWO COURSES
A15 EVOLUTIONARY STAGES OF STUDENT PROJECTS
A16 REFLECTIVE ENGAGEMENTS
A17 FEEDBACK FROM THE CROSS-LEVEL MENTORS ON INFUSION OF SOME PERVERSIVE TOPICS IN FOUNDATION COURSES