

1. Record Nr.	UNISA996495562003316
Titolo	Accelerating digital transformation : 10 years of Software Center // Jan Bosch [and four others] editors
Pubbl/distr/stampa	Cham, Switzerland : , : Springer, , [2022] ©2022
ISBN	3-031-10873-6
Descrizione fisica	1 online resource (447 pages)
Disciplina	005.1
Soggetti	Software engineering
Lingua di pubblicazione	Inglese
Formato	Materiale a stampa
Livello bibliografico	Monografia
Nota di bibliografia	Includes bibliographical references.
Nota di contenuto	Intro -- Accelerating Digital Transformation -- Foreword -- Preface -- Contents -- Part I Continuous Delivery -- Introduction to the Continuous Delivery Theme -- 1 Climbing the Stairway to Heaven -- 1.1 Introduction -- 1.2 From Agile Development to Continuous Deployment of Software -- 1.2.1 Traditional Development -- 1.2.2 Agile R&D Organization -- 1.2.3 Continuous Integration -- 1.2.4 Continuous Deployment -- 1.2.5 R&D as an 'Experiment System' -- 1.2.6 Summary -- 1.3 Research Approach -- 1.3.1 Research Sites -- 1.3.2 Research Method -- 1.4 Data Collection and Analysis -- 1.5 Validity and Generalizability of Results -- 1.6 Case Study Findings -- 1.6.1 Company A -- 1.6.2 Company B -- 1.6.3 Company C -- 1.6.4 Company D -- 1.7 Climbing the Stairway to Heaven -- 1.7.1 From Traditional to Agile R&D -- 1.7.2 From Agile R&D to Continuous Integration -- 1.7.3 From Continuous Integration to Continuous Deployment -- 1.7.4 From Continuous Deployment to Innovation System -- 1.8 Conclusions -- 2 Modeling Continuous Integration Practice Differences in Industry Software Development -- 2.1 Introduction -- 2.2 Research Method -- 2.2.1 Systematic Review -- 2.2.2 Analysis of Literature -- 2.2.3 Proposing a Model -- 2.3 Statement Clusters -- 2.3.1 Culled Clusters -- 2.3.1.1 Build Version Selection -- 2.3.1.2 Component Dependency Versioning -- 2.3.1.3 Fault Frequency -- 2.3.1.4 Fault Responsibility -- 2.3.1.5 Lifecycle Phasing -- 2.3.1.6 Process Management -- 2.3.2 Preserved Clusters --

2.3.2.1 Build Duration -- 2.3.2.2 Build Frequency -- 2.3.2.3 Build Triggering -- 2.3.2.4 Definition of Failure and Success -- 2.3.2.5 Fault Duration -- 2.3.2.6 Fault Handling -- 2.3.2.7 Integration Frequency -- 2.3.2.8 Integration on Broken Builds -- 2.3.2.9 Integration Serialization and Batching -- 2.3.2.10 Integration Target -- 2.3.2.11 Modularization.

2.3.2.12 Pre-integration Procedure -- 2.3.2.13 Scope -- 2.3.2.14 Status Communication -- 2.3.2.15 Test Separation -- 2.3.2.16 Testing of New Functionality -- 2.4 A Descriptive Model -- 2.4.1 Integration Flow Anatomy -- 2.4.2 Node Attributes -- 2.4.2.1 Scope Attributes -- 2.4.2.2 Build Characteristics Attributes -- 2.4.2.3 Result Handling Attributes -- 2.4.2.4 Input Node Attributes -- 2.4.3 Attribute Selection and Constraints -- 2.5 Illustrative Case Study -- 2.5.1 Project A -- 2.5.2 Sketching the Integration Flow Anatomy -- 2.5.3 Determining the Input Node Attributes -- 2.5.4 Determining the Activity Node Attributes -- 2.6 Conclusion -- 2.6.1 Disagreements in Related Work -- 2.6.2 Model Proposal -- 2.6.3 Model Validation -- 2.6.4 Open Questions for Further Research -- 2.6.4.1 Correlations Between Differences in Practice and Differences in Experience -- 2.6.4.2 Contextual Differences -- 2.6.4.3 Internal Constraints and Correlations of the Model -- 3 Efficient and Effective Exploratory Testing of Large-Scale Software Systems -- 3.1 Introduction -- 3.1.1 Background and Related Work -- 3.1.2 Research Question -- 3.1.3 Contribution -- 3.2 Research Method -- 3.2.1 Overview of the Research Method -- 3.2.2 Reviewing Literature -- 3.2.3 Identifying the Key Factors -- 3.2.4 Confirming the Key Factors -- 3.2.5 Development of the ExET Model -- 3.2.6 Validation of the ExET Model -- 3.3 Reviewing Literature -- 3.3.1 Criteria for the Literature Review -- 3.3.2 Results from the Literature Review -- 3.4 Identifying the Key Factors -- 3.4.1 Background Information -- 3.4.2 Key Factors for Efficient and Effective Exploratory Testing -- 3.5 Confirming the Key Factors -- 3.5.1 Follow-Up Interviews -- 3.5.2 Cross-Company Workshop -- 3.6 The ExET Model -- 3.6.1 A Description of the ExET Model -- 3.6.2 How to Use the Model -- 3.7 Validation of the ExET Model.

3.7.1 The Five Validation Cases -- 3.7.2 Validation Cross-Company Workshop -- 3.7.3 Summary and Analysis of the Validation -- 3.8 Threats to Validity -- 3.8.1 Threats to Construct Validity -- 3.8.2 Threats to Internal Validity -- 3.8.3 Threats to External Validity -- 3.9 Conclusion and Further Work -- 3.9.1 Conclusion -- 3.9.2 Further Work -- Part II Continuous Architecture -- Introduction to the Continuous Architecture Theme -- 4 Technical Debt Tracking: Current State of Practice: A Survey and Multiple Case Study in 15 Large Organizations -- 4.1 Introduction -- 4.2 Methodology -- 4.2.1 Survey -- 4.2.1.1 Survey Data Collection -- 4.2.1.2 Survey Data Analysis -- 4.2.2 Multiple Case Study -- 4.2.2.1 Interviews -- 4.2.2.2 Document Analysis -- 4.3 Results -- 4.3.1 Demographics and Background of the Respondents -- 4.3.2 Estimation of Management Cost of TD (RQ1) -- 4.3.3 Familiarity with the Term "Technical Debt" (RQ2) -- 4.3.4 Awareness of Technical Debt Present in the System (RQ3 and RQ5) -- 4.3.5 Tracking Technical Debt (RQ4) -- 4.3.6 Influence of the Background of Respondents on the Management of TD (RQ6) -- 4.3.7 Tools Used to Track Technical Debt (RQ7) -- 4.3.8 Why and How Do Companies Start Tracking TD? (RQ3) -- 4.3.8.1 Motivation for Start of TD Tracking -- 4.3.8.2 Preparation of the Tracking Process -- 4.3.9 What Are the Benefits and Challenges of Tracking TD? (RQ4) -- 4.3.9.1 Benefits -- 4.3.9.2 Challenges -- 4.3.10 Strategic Adoption Strategy -- 4.4 Discussion -- 4.4.1 Current State of Practice of Tracking TD and Implications for Practitioners and Researchers -- 4.4.2 Related Work --

4.4.3 Limitations and Threats to Validity -- 4.5 Conclusion -- 5
Expectations and Challenges from Scaling Agile in Mechatronics-Driven
Companies - A Comparative Case Study -- 5.1 Introduction -- 5.1.1
Problem Domain and Motivation -- 5.1.2 Research Goal.
5.1.3 Contributions and Scope -- 5.1.4 Structure of the Article -- 5.2
Related Work -- 5.3 Comparative Case Study Design -- 5.3.1 Research
Questions -- 5.3.2 Case and Subjects Selection -- 5.3.3 Data
Collection Procedure -- 5.3.3.1 Individual On-Site Workshops --
5.3.3.2 Survey -- 5.3.3.3 Joint Workshop -- 5.3.4 Analysis Procedure
-- 5.3.4.1 Individual On-Site Workshops -- 5.3.4.2 Survey -- 5.3.4.3
Joint Workshop with External Agile Expert -- 5.3.5 Validity Procedure
-- 5.4 Results -- 5.4.1 Threats to Validity -- 5.5 Conclusion and Future
Work -- 6 Lightweight Consistency Checking for Agile Model-Based
Development in Practice -- 6.1 Introduction -- 6.2 Scope -- 6.2.1
Industrial Context of Consistency Checking -- 6.2.2 Requirements --
6.3 Our Consistency-Checking Approach -- 6.3.1 Language
Consistency Mapping -- 6.3.2 Model Consistency Mapping -- 6.3.3
Continuous Integration Pipeline -- 6.4 Proof of Concept -- 6.4.1
Language Consistency Mapping -- 6.4.2 A Consistency Checking Tool
-- 6.5 Discussion -- 6.6 Related Work -- 6.7 Conclusions and Future
Work -- Part III Metrics -- Introduction to the Metrics Theme -- 1
Introduction -- 2 Software Metrics in 2010, 2020, and Beyond -- 3
Industrial Tools and Methods Developed in Software Center -- 4 The
Future -- 7 MESRAM - A Method for Assessing Robustness of
Measurement Programs in Large Software Development Organizations
and Its Industrial Evaluation -- 7.1 Introduction -- 7.2 Related Work --
7.3 Measurement Programs -- 7.3.1 Defining and Using Measures --
7.3.2 Implementation of Measurement Programs -- 7.3.2.1
Measurement Systems -- 7.3.2.2 Measurement Infrastructure --
7.3.2.3 Measurement Organization -- 7.3.3 Measurement Program
Model -- 7.4 MeSRAM - Measurement System Robustness Assessment
Method -- 7.4.1 Developing MeSRAM Using Action Research -- 7.4.1.1
Action Research Cycle 1: Initial Development.
7.4.1.2 Action Research Cycle 2: Improvement and Application -- 7.4.2
Robustness Model -- 7.4.2.1 Metrics Organization -- 7.4.2.2 Metrics
Infrastructure -- 7.4.2.3 Metrics Used -- 7.4.2.4 Decision Support --
7.4.2.5 Organizational Metrics Maturity -- 7.4.2.6 Collaboration with
Academia -- 7.4.2.7 External Collaborations -- 7.4.3 Assessment
Method -- 7.5 Evaluation of MeSRAM in Industry -- 7.5.1 Design of the
Evaluation -- 7.5.2 Results of the Pilot Study at Company A -- 7.5.3
Results of Assessment -- 7.5.4 Feedback from the Companies -- 7.5.5
Identified Improvements of the Measurement Programs -- 7.6 Analysis
and Interpretation -- 7.6.1 Analysis per Company -- 7.6.1.1 Company
A -- 7.6.1.2 Company B -- 7.6.1.3 Company C -- 7.6.1.4 Company D
-- 7.6.1.5 Company E -- 7.6.1.6 Company F -- 7.6.1.7 Company G --
7.6.1.8 Summary -- 7.6.2 Differences per Process Model -- 7.6.3
Differences per Product Type -- 7.6.4 Differences per Development
Type -- 7.7 Threats to Validity -- 7.8 Conclusions -- 8 Recognizing
Lines of Code Violating Company-Specific Coding Guidelines Using
Machine Learning -- 8.1 Introduction -- 8.2 Related Work -- 8.2.1
Comparison Between Tools -- 8.2.2 Machine Learning for Static Code
Analysis -- 8.2.3 Machine Learning for Code-Smell Detection -- 8.2.4
Summary -- 8.3 The CCFlex Tool -- 8.3.1 Architecture -- 8.3.2
Feature-Extraction Filters -- 8.3.3 Classification Algorithms -- 8.3.4
Active Learning -- 8.4 Research Methodology and Design -- 8.5
Execution and Results -- 8.5.1 Action Research Cycle 1 - What Coding
Guidelines Are Used by Our Industrial Partners? -- 8.5.1.1 Cycle Goal
and Research Procedure -- 8.5.1.2 Cycle Execution and Results --

8.5.2 Action Research Cycle 2 - Selecting a Tool Capable of Recognizing Code Guidelines Violations of Our Partners -- 8.5.2.1 Cycle Goal and Research Procedure -- 8.5.2.2 Cycle Execution and Results.

8.5.3 Action Research Cycle 3 - How Can We Recognize the Violations Provided by the Industrial Partners?.
