Human Computer Interaction

- Human-Computer Interaction, Alan Dix, Janet Finlay, Gregory D. Abowd

	Requirements Engineering for Software and Systems
	Introduction to Requirements Engineering
	Motivation
	What is Requirements Engineering?
	You Probably Don't Do Enough Requirement Engineering
	What are Requirements?

UX Design

- The Elements of User Experience: User-Centered Design for The Web and Beyond, Jesse James Garrett
- Killer UX Design, Jodie Moule
- The Guide to UX Design Process and Documentation, Dominik Pacholczyk
- About Face: The Essentials of Interaction Design, Alan Cooper, Robert Reimann, David Cronin

	Requirements Engineering for Software and Systems
	Introduction to Requirements Engineering
	Motivation
	What is Requirements Engineering?
	You Probably Don't Do Enough Requirement Engineering
	What are Requirements?

UI Design

- The Fundamentals of Graphics Design, Gavin Ambrose
- The Non-Designer's Design Book: Design and Typographic Principles for the Visual Novice, Robin Williams
- Designing the User Interface: Strategies for Effective Human-Computer Interaction, Shneiderman, Plaisant
- Interface Design, An Introduction to Visual Communication in UI Design
- Designing Interface: Patterns for Effective Interaction Design, Jenifer Tidwell

	Requirements Engineering for Software and Systems
	Introduction to Requirements Engineering
	Motivation
	What is Requirements Engineering?
	You Probably Don't Do Enough Requirement Engineering
	What are Requirements?

System Analysis and Design

- Modern System Analysis and Design, Joseph S.Valacich
- Systems Analysis and Design with UML Version 2.0: An Object-Oriented Approach, Dennis, Wixom

	Requirements Engineering for Software and Systems
	Introduction to Requirements Engineering
	Motivation
	What is Requirements Engineering?
	You Probably Don't Do Enough Requirement Engineering
	What are Requirements?

Software Requirements

- Requirements Engineering for Software and Systems, Phillip A. Laplante

	Requirements Engineering for Software and Systems
	Introduction to Requirements Engineering
	Motivation
	What is Requirements Engineering?
	You Probably Don't Do Enough Requirement Engineering
	What are Requirements?
	Requirements Engineering Activities
	Bodies of Knowledge
	The Requirements Engineer
	Requirements Engineer Roles
	Role of the Customer
	Problems with Traditional Requirements Engineering
	Difficulties in Enveloping System Behavior
	Exercises
	References
	Preparing for Requirements Elicitation

	Production Mission Statement
	Encounter with a Customer
	Identifying the System Boundaries
	Stakeholders
	Customer Wants and Needs
	Why Do Customers Change Their Minds?
	Stakeholder Prioritization
	Communicating with Customers and Other Stakeholders
	Stakeholder Negotiations
	Uncovering Stakeholder Goals
	Exercises
	References
	Requirements Elicitation
	Introduction
	Preparing for Requirements Elicitation
	Elicitation Techniques Survey – Topics 6
	Elicitation Techniques Survey – Topics 6
	Elicitation Techniques Survey – Topics 6
	Elicitation Techniques Survey – Topics 4
	Eliciting Nonfunctional Requirements
	Elicitation Summary
	Eliciting Hazards
	Exercise
	References
	Writing the Requirements Document
	Requirements Agreement and Analysis
	Requirements Representation
	ISO/IEC/IEEE Standard 29148
	UML/SysML
	The Requirement Document – Topics ½
	The Requirement Document – Topics ½
	Behavioral Specifications
	Best Practices and Recommendations
	Exercises
	Reference
	Requirements Risk Management
	What is Requirements Risk Management?
	Requirements Validation and Verification – Topics ½
	Requirements Validation and Verification – Topics ½
	Standards for V&V – Topics ½
	Standards for V&V – Topics ½
	NASA Requirements Testing – Topics ½
	NASA Requirements Testing – Topics ½
	Exercises
	Reference
	Formal Methods
	Motivation
	What Are Formal Methods?
	Examples – Topics 1/3
	Examples – Topics 1/3
	Examples – Topics 1/3

	Use Case
	IBM DOORS Requirements Management Tool

Software Metrics

- Applied Software Measurement – Global Analysis of Productivity and Quality, Capers Jones

	Applied Software Measurement
	Introduction
	Applied Software Measurement
	Planning and Estimation
	Management and Technical Staffs
	Organization Structures
	Methodologies and Tools
	The Office Environment
	Reusability
	The Essential Aspects of Applied Software Measurement
	What Do Companies Measure?
	Benchmarks and Industry Measures
	Measurement and The Software Life Cycle
	The Structure of a Full Applied Software Measurement System
	The Sociology of Software Measurement

	The Sociology of Data Confidentiality
	The Sociology of Using Data for Staff Performance Tragets
	The Sociology of Measuring One-Person Projects
	The Sociology of MIS vs. Systems Software
	Justifying and Building an Applied Software Measurement Function
	Applied Software Measurement and Future Progress
	Suggested Readings
	Additional Readings On Software Measurement and Metrics
	The History and Evolution of Software Metrics
	Evolution of the Software Industry and Evolution of Softwre Measuremetns
	The Cons of Counting Function Point Metrics
	The Paradox of Reversed Productivity of High-Level Languages
	The Varieties of Functional Metrics Circa 2008
	Variations in Application Size and Productivity Rates
	Future Technical Developments in Functional Metrics
	Summary of and Conclusion About Functional Metrics
	Software Measures and Metrics Not Based On Function Points
	Suggested Readings on Measures and Metrics
	United Stated Averages for Software Productivity and Quality
	Sources of Possible Errors in the Data
	Significant Software Technology Changes Between 1990 and 2008
	Changes in the Structure, Format, and Contents of the Third Edition
	Variations in Software Development Practices Among Seven Sub-Industries
	Ranges, Averages, and Variances in Software Productivity
	The Impact of Technology ON Software Productivity and Quality Levels
	Technology Warnings and Counter Indications
	Using Function Point Metrics to Set “Best in Class” Targets
	The Mechanics of Measurement: Building a Baseline
	Software Assessments
	Software Baselines
	Software Benchmarks
	What a Baseline Analysis Covers
	Developing or Acquiring a Baseline Data Collection Instrument
	Administering the Data Collection Questionnaire
	Analysis and Aggregation of the Baseline Data
	Suggested Readings
	Additional Readings
	Measuring Software Quality and User Satisfaction
	New Quality Information Since the Earlier Editions
	Quality Control and International Competition
	Defining Quality for Measurement and Estimation
	Five Steps to Software Quality Control
	Software Quality Control in the United States
	Measuring Software Defect Removal
	Measuring Defect Removal Efficiency
	Finding and Eliminating Error-Prone Modules
	Using Metrics to Evaluate Test-Case Coverage
	Using Metrics for Reliability Prediction
	Measuring the Costs of Defect Removal
	Evaluating Defect Prevention Methods
	Measuring Customer-Reported Defects

	Measuring Invalid Defects, Duplicate Defects, and Special Case
	Measuring User Satisfaction
	Combining User Satisfaction and Defect Data
	Summary and Conclusions
	Reading List
	Suggested Readings
	Additional References on Software Quality and Quality Measurements
	Measurements, Metrics, and Industry Leadership
	What Do Companies Measure?
	Measures and Metrics of Industry Leaders
	Measures, Metrics, and Innovation
	Measurements, Metrics, and Outsource Litigation
	Measurements, Metrics, and Behavioral Changes
	Topics Outside the Scope of Current Measurements
	Cautions Against Simplistic and Hazardous Measures and Metrics
	Commercial Software Measurement Tools
	Summary and Conclusions
	Suggested Readings on Measurement and Metrics
	Summary of Problems in Software Measurement
	Synthetic vs. Natural Metrics
	Ambiguity in Defining the Nature, Scope, Class, and Type of Software
	Ambiguity in Defining and Measuring the Activities and Tasks of Software Projects
	False Advertising and Fraudulent Productivity Claims
	The Absence of Project Demographic and Occupation Group Measurement
	Ambiguity in the Span of Control and Organizational Measurements
	The Missing Link of Measurement: When Do Projects Start?
	Ambiguity in Measuring Milestones, Schedules, Overlap, and Schedule Slippage
	Problems with Overlapping Activities
	Leakage from Software Project Resource Tracking Data
	Ambiguity in Standard Time Metrics
	Inadequate Undergraduate and Graduate Training in Software Measurement and Metrics
	Inadequate Standards for Software Measurement
	Lack of Standardization of “Lines of Source Code” Metrics
	The Hazards and Problems of Ratios and Percentages
	Ambiguity in Measuring Development of Delivery Productivity
	Ambiguity in Measuring Complexity
	Ambiguity in Functional Metrics
	Ambiguity in Quality Metrics
	Ambiguity with the Defects Per KLOC Metric
	Ambiguity with the Cost per Defect Metric
	Failure to Measure Defect Potentials and Defect Removal Efficiency
	The Problems of Measuring the Impact of “Soft” Factors
	Problems in Measuring Software Value
	Lack of Effective Measurement and Metrics Automation
	Social and Political Resistance to Software Measurements
	Ambiguity in Software Measurement and Metrics Terminology
	Failure to Use Metrics for Establishing Goals and Targets
	Summary and Conclusions
	Suggested Readings
	Additional References on Software Measurements
	Appendix: Rules for Counting Procedural Source Code

	Project Source Code Counting Rules
	General Rules for Counting Code Within Applications
	Examples of the SPR Source Code Counting Rules
	Software Productivity Research COBOL-Counting Rules