LECTURE NOTES

LN 01/17: Software Engineering Processes...1

- Hooker's SW Engineering principles 1
- SWE fallacies 1
- Ethical issues 1

LN 01/22: Requirements Engineering 2

- Dynamic and Practice perspectives 2
- Good requirement properties 2
- Types of requirements 3
- Requirements negotiation 4
- Requirements validation 4
- Requirements management 4
- Cost of fixing defects 4

LN 01/24: Software Processes...5

- Construction vs. Engineering 5
- Extreme Programming phases 6
- Spike solutions 6
- XP values 6
- XP at scale 7
- Downsides of agile 7
- System engineering 8
- Hayley-Pirbhai modeling 8
- System modeling 9
- Data modeling 10
- Use case modeling 10

LN 01/31: Modeling the Problem..11

- Reasons for writing use cases 11
- Sequence diagrams 11
- Why architectural design? 12
- What we want of architectural design? 12
- Patterns, Architectural: List 12

LN 02/05: Architecture...13

- Multitier Client-Server 13
- Distributed components 13
- Peer-to-peer 13
- Software as a Service (SaaS) 13
- Distributed architectures Pros/Cons 13
- Service Oriented Architecture (SOA) 14
    - SOA process
    - SOA standards (SOAP, WSDL, WS-BPEL)
- RESTful 14
- Component Design 15
- Design concepts 15
- Design patterns 16
    - Design patterns categories 16

LN: 02/07: QA and Configuration Management...........................17

- Class Design 17
    - Different mechanisms for implementing
    - When to create a class? 17
    - When no to create a class? 17
    - Design principles for class-level design 17
    - Leskov substitution principle 17
- Package design principles 17
- Quality Management 18
- Plan-driven QA 19
- Agile QA 19

LN: 02/14: Testing...20

- What to test? 20
- Error statistics 20
- System testing 20
- Verification and Validation 20
- Fault/Failure/Error 21
- Drivers/Stubs 21
- Test-driven development (TDD) 21
- Issues with testing 21
- Tinderboxes 21
- Whitebox vs. blackbox 22
- Equivalence partitioning 22
- Black box testing advantages 23
- Budgeting for testing 23

LN 02/21: Project Management...23

- Build process 23
- CM questions 23
- Baseline 24
- Version Control types 24
- Version Control 24
- Project Management Danger Signs 24
- People Management 25-26
- Personalities division 26
- Project Management, key parameters 26
- Project plan components 27
- Risk Management 27
    - Risk Estimation 27-28
- Project Estimation 28
- Late Projects 28

LN 02/28: Dependability and Security..29

- Dependability parameters 29
- Dependability terminology 29
- Safety terminology 29
- Security terminology 29
- Dependability components 29
- Dependability requirements 30
- Faults, dealing with 30
- Dependable SW processes 30
- Dependable programming practices 30
- Dependability hindering practices 30
- Security 31
- Security is harder than dependability 31
- Security design guidelines 31
- Security architecture guidelines 31

LN 03/05: Software Evolution..32

- CBSE-Component Based Software Engineering 32
    - Package assessment 32
- Reuse business considerations 33
- Statistics to changes to SW 33
- Debugging 33
- Debugging procedures 33
- Debugging, Questions while 34
- Debugging heuristics 34
- Refactoring methods 34
- Tuning 34
- Tuning, Alternatives to 34

LN 03/07: Tuning, Tools, Formal Methods...36

- Tuning tactics 36
- Formal methods ideas (Invariant/Pre-, Post-conditions)

LN: 03/12: Formal Specs, Comments, Documentation, UI............39

- Formal Specs 38
- Z language 38
- Formal methods, downsides 38
- Comments and Documentation 39

- Comments, How to avoid 39
- Comment, When to 39
- Comment problems 39
- Documentation 39
- Process Documentation 39
- Product Documentation 40
- Usability 41
- Accessibility 41

LN 03/14: UI Design…………………………………………………………………….....42

- User Interface 40
- UI, Common Issues 40
- Web UI 41
- UI, Mobile App 41
- Usability testing 42
- Reuse (Pros, Cons) 42
- Reuse techniques 42
- SW Process Improvement 43
- BPR (Business Process Reengineering) 43
- SW Reengineering, Technical methods 43
- Schorsch’s immaturity levels 43
- Software and Law 43
- Commercial law pertaining to SW 44

McConnell

MC Ch 03: Upstream Prerequisites..47

- Incomplete preparation, causes 47
- Prerequisites before construction, why? 47
- Requirements, why? 49
- Requirements changes, dealing with 49
- Good requirements specification checklist 49
- Architectural prerequisites 50
- Architectural components 50
- Upstream prerequisites, how much time to spend? 52

MC Ch 04: Key Construction Decisions...53

- Technology wave 53
- Major construction practices 53
- Managing complexity, importance 55
- Attacking complexity 55
- Design, desirable characteristics 55
- Design Levels 56
- Levels at which design should be specified 56
- Subsystems 56
- Heuristics as design building blocks 57
- Information Hiding, importance 57
- Identifying areas likely to change 57
  - Isolating unstable areas 58
  - Anticipating degrees of change 58
  - Keep coupling loose 58
  - Coupling, Types 58
- Design patterns 59
- Design patterns, benefits 59
- Heuristics, Other 59
- Design practices 59
- Collaborative design 61
- Design, how much is enough? 61
- Popular methodologies 61

MC Ch 06: Working Classes...62

- Abstract Data Type 62
- Class interfaces 62
- Encapsulation heuristics 63
- Issues of class design/implementation 63
- Containment 64
- Inheritance 64
- Why create a class? 65
- Inheritance vs Containment 65
- Classes to avoid 66
- Classes, language-specific concerns 66
- Packages 66

MC Ch 20: Software Quality Landscape...67

- External and Internal quality characteristics 67
- SW Quality elements 67
- SW Quality principles 68

MC Ch 22: Developer Testing..69

- Testing, why is it hard? 69
- Testing, how much time to spend? 69
- Testing, what to test? 69
- Data flow testing 70
- Equivalence partitioning 70
- Improving your tests 72
- Test records 73

MC Ch 23: Debugging...74

- Learning through errors 74
- Bad debugging 74
- Scientific approach 74
- Debugging, brute-force approach 75
- Syntax errors 75
- Fixing defects 76
- Debugging tools 76

MC Ch 24: Refactoring...77

- Refactoring, when to? 77
- Data-level refactoring 78
- Statement-level refactoring 79
- Routine-level refactoring 79
- Class refactoring 79
- Class interface refactoring 80
- System-Level refactoring 80
- Refactoring safely 81
- Bad times to refactor 81
- Refactoring strategies 81

MC Ch 25: Code Tuning Strategies..82

- Efficiency viewpoints 82
- Code tuning 83
- Inefficiency, common sources 83
- Measurement 84

MC Ch 26: Code-Tuning Techniques..85

MC Ch 27: Effects of Program Size on Construction......................85

MC Ch 28: Managing Construction..85

- Encouraging good coding 88
- Configuration Management 88
- Requirements and design changes 88
- Controlling design changes 89
- Version Control benefits 89
- Scheduling construction 89
  - Estimation approaches 89
- Factors that influence SW Project effort 91
- Resource control to meet schedule 92
- Measurement 92
- Programmers as People 92
- Managing your manager 93

MC Ch 29: Integration..94

- Integration quality importance 94
- Phased integration 94
- Incremental integration 94
  - Advantages 95
- Top-down vs bottom-up integration 95
- Daily Build 96
- Smoke tests 96

MC Ch 29: Programming tools...98

MC Ch 31: Layout and Style...101

- Good layout objectives 101

MC Ch 32: Self-Documenting Code..104

- Self-documenting code checklist 104
- Comments, types of 105
- Effective commenting 105
- Commenting techniques 106

Sommerville

SV Ch 01: Introduction..........109

- Kinds of SW products 109
- Product characteristics 109
- Fundamental activities of SW processes 109
- Issues that affect many types of SW 109
- Ethical principles 110
- Types of software systems 110

SV Ch 02: Software Processes..........111

- The Waterfall model 112
- Incremental development 112
- Reuse-oriented SW engineering 113
- Requirements engineering 113
- Design and implementation 114
- Validation 114
- Change 115
- Prototyping 115
- Incremental delivery 115
    - Advantages 116
    - Bohm's spiral 116
- RUP (Rational Unified Process) 117

SV Ch 03: Agile Software Development..........119

- What to consider before deciding on agile 119
- Plan-based vs Agile 120
- XP (Extreme Programming) 121
- XP Testing 121
- Pair programming 121
    - Advantages 122
- Scrum: Agile project management 122
    - Advantages 122
- XP, Scaling 122
    - Scaling up vs scaling out 123
- Difficulties in introducing agile 123

SV Ch 04: Requirements Engineering..........124

- Requirements: IEEE Standard 125
- Closed vs Open interviews 126
- Use case example 127
- Change management process 127

SV Ch 05: System Modeling..........129

- Perspectives of models 129
- How are models used? 129
- Context models 129
- Interaction models 130
- Model-driven engineering (MDE) Pros/Cons 132
- Model-driven architecture (MDA) 132

SV Ch 06: Architectural Design..........134

- Questions concering any system 134
- Architectural views 134
    - 4+1 view model of SW architecture 134
- Architectural patterns 135
- Application architectures 136
    - How to use them? 136

SV Ch 07: Design and Implementation..........138

- Design patterns, essential elements 138
- Issues with implementation 139
- Open source development 139
    - Open source corporate considerations 140

SV Ch 08: Software Testing..........141

- Inspection vs testing 141
- Development testing 141
- Unit testing 141
    - How to choose test cases? 142
    - Testing boundary values 142
    - Component testing 142
    - System testing 142
- Test-Driven Development (TDD) 143
- Release testing 144
- Performance testing 144
- User testing 144
- Acceptance testing 144

SV Ch 09: Software Evolution..........145

- Evolution processes 145
- Evolution dynamics 145
    - Lehman's Laws 145
- Software maintenance 146
- Maintainability assessment 147
- Software reengineering 147
- Refactoring as preventative maintenance 147
    - Fowler's bad smells 147
- Legacy system management 148
    - Evolution options 148
    - Types of legacy systems 148
    - Environmental system assessment 148
    - Supplier stability 148
    - Technical system assessment 149
    - Quality assessment data 149

SV Ch 10: Sociotechnical Systems..........150

- Layers of sociotechnical system stack 150
- Complex systems 150
- Security and dependability factors 151
- Emergent system properties 151
- System engineering 151
- Procurement 152
- System development activities 152
- Human error 153
- System evolution 153

SV Ch 11: Dependability and Security..........154

- Dependable system properties
- Availability and reliability 154
- Improving reliability 155
- Safety 155
- Reliability vs safety 155
- Safety terminology 155
- Minimizing accident consequences 155
- Security 156

SV Ch 12: Dependability and Security Specification..........157

- Risk-driven requirements specification 157
- Safety specification 157
- Risk classification 157
- Reliability specification 158
- Non-functional reliability requirements 158
- Functional reliability requirements 158
- Security specification 159
- Formal specification 160
    - Advantages 160
    - Disadvantages 160

SV Ch 13: Dependability Engineering..........161

- Redundancy and diversity 161
- Dependable processes 161
- Dependable system architectures 161

- N-version programming 162
- Software diversity 162
- Dependable programming practices 162

SV Ch 14: Security Engineering……………………………………………………164

- Security risk management 164
- Security threats 164
- Design for security 165
- Design Guidelines 165
- Design for deployment 166
- System survivability 167

SV Ch 15: Dependability and Security Assurance…………………….168

- Verification and formal methods 168
- Model checking 168
- Automatic static analysis 169
- Reliability testing (PODOF/ROCOF) 170
    - ○ Measuring system reliability 170
- Security testing 170
- Security checklist 171
- Process assurance 171
- Processes for safety assurance 171
- Identifying responsible engineers, importance 172
- Safety and dependability cases 172
- Structured arguments 172

SV Ch 16: Software Reuse……………………………………………………….174

- Frameworks 174
- Web Frameworks 174
- Product ilnes 175
- COTS product reuse 175
    - ○ Downsides of COTS 176

SV Ch 17: Component-Based Software Engineering…………………177

- Characteristics of reusable components 177
- Component models 178
- CBSE processes 178
- Component acquisition, management, certification 178
- CBSE for reuse 179
- Improving reusability via component changes 179
- Issues with not handling exception 179
- CBSE vs nominal software engineering 179
- Types of interface incompatibilities 180

SV Ch 18: Distributed Software Engineering……………………………181

- Issues 181
- Interaction models 182
- Client-server computing 182
- Architectural patterns for distributed systems 182
- Software as a Service (SaaS) 183
- SaaS vs SOA 183
- Scalable software development guidelines 183

SV Ch 19: Service-Oriented Architecture…………………………………184

- SOA Standards (SOAP, WSDL, WS-BPEL) 184
- RESTful services 184
- Service engineering 184
    - ○ Types of services 184
- Determining whether a service is general/reusable 185
- Service interface design 185
- Software development with services 185
  Service construction by composition 185

SV Ch 20: Embedded Software……………………………………………….187

- Embedded systems vs. other software 187
- Embedded system design 187
- Real-time software design process 187
- Real-time Operating Systems (RTOS) 188

SV Ch 22: Project Management……………………………………………….190

- Responsibilities of project managers 190
- Risk management 190
- Categories of risk 190
- Risk management processes 190
- Managing people 191
- Personality types 192
- Teamwork 192
- Organization questions for project managers 192

SV Ch 23: Project Planning……………………………………………………….194

- What should a plan contain? 194
- Planning process 194
- Project scheduling 195
- Estimation techniques 196
- COCOMO II Model 196
- Application-composition model 196
- Early design model 197
- Reuse model 197
- Post-architectural level 198

SV Ch 24: Quality Management………………………………………………199

- Concerns 199
- Quality plan structure 199
- Software standards 200
- ISO 9001 standard 200
- Reviews and Inspection 201
- Measurement and metrics 201

SV Ch 25: Configuration Management…………………………………….203

- Activities 203
- Version management 203
- System building 204
- Build system features 204
- Continuous integration 204
    - ○ Issues 205
- Daily build 205
- Release management 205
    - ○ What should a release include? 206

SV Ch 26: Process Improvement……………………………………………….206

- The Process improvement process 206
- Process measurement 207
- GQM (Goal-Question-Metric paradigm) 207
- Process analysis 207
- Process analysis techniques 208
- Questions about the formal process model 209
- Process change 210
- CMMI process improvement framework 210
- Staged CMMI 212
- Continuous CMMI

SV Ch 27: Formal Specification…………………………………………………214

date	McConnell	Sommerville	Topics
01-08			introduction
01-10	*(Read homework.)*		initial project meeting A
01-12			*requirements gathering (in discussion)*
01-17			initial project meeting B
01-19			*initial project organization (in discussion)*
01-22	1, 3–3.4	1, 4	requirements engineering
01-24	2, 21	2–3	software process models
01-26			*requirements homework due on CCLE*
01-29		5, 19, 20	system modeling and engineering
01-31	3.5	6, 17, 28	architectural design
02-05	4, 5, 6	7, 16, 18	component-level design
02-07	20	24, 25	quality and change management
02-12			*midterm (in lecture)*
02-14	22, 29	8	testing
02-21	28	22	project management
02-23			*midterm presentations (in discussion)*
02-26	3.6, 27	23	project planning
02-28		10–14	dependability and security
03-05	23–26, 31	9	debugging, refactoring, and tuning
03-07	30	27	tools and formal methods
03-12	32	29, 30	user interface and documentation
03-13			*project final report due (on CCLE)*
03-14	33–35	15, 26	reuse and process improvement
03-16			*final presentations (in discussion)*