

ICS225: Software Processes

Susan Elliott Sim
January 10, 2005

Overview

- Course Description
- Course Organization
- Software Process: A Roadmap

Week 1, Monday Lecture Slide 2

Software Process

- Methods and technologies used to assess, support, and improve software development activities.
- “A software process can be defined as the coherent set of policies, organizational structures, technologies, procedures, and artifacts that are needed to conceive, develop, deploy, and maintain a software product.”
- Hard to explain to people outside of software engineering

Week 1, Monday Lecture Slide 3

Sub-Areas in Software Process

- Modeling
- Simulation
- Metrics
 - Definition, deployment, evaluation
- Process Improvement
- Standardization
- Quality Assurance
 - Inspections
- Lifecycle
- Methods
 - Cleanroom, PSP
- Workflow

Week 1, Monday Lecture Slide 4

Gaps and Tensions

- Prescription vs. Description
- Understanding vs. Improvement
- Professional publications vs. research literature
 - Examples of former: Fred Brooks, Tom DeMarco, Steve McConnell, Martin Fowler
 - Examples of latter: readings in this course, journal and conference papers
- Tools vs. Methods
- Social vs. Formal

Week 1, Monday Lecture Slide 5

Course Organization

- Research-oriented
- Problem-based learning
- Readings mostly from Phase II List

Week 1, Monday Lecture Slide 6

Evaluation

- | | |
|--------------------|-----|
| • Term Project | 70% |
| • Project Proposal | 10% |
| • Debate | 10% |
| • Participation | 10% |

Week 1, Monday Lecture Slide 7

Software Process: A Roadmap

- A collection of roadmap was published as part of ICSE2000
 - Each roadmap was written by a leader in a particular research area
 - Survey from a particular point of view
 - High variability in quality
 - Has now become a staple in graduate courses
- Alfonso Fuggetta wrote the one for software process
 - Also a member of ISR

Week 1, Monday Lecture Slide 8

Definition

- “A software process can be defined as the coherent set of policies, organizational structures, technologies, procedures, and artifacts that are needed to conceive, develop, deploy, and maintain a software product.”
- Software is complex (see Brooks)
- Software is built by people
- Good process produces quality software

Week 1, Monday Lecture Slide 9

Contributing Areas

- Software development technology
- Software development methods and techniques
- Organizational behavior
- Business administration and economics

Week 1, Monday Lecture Slide 10

Historical Research Results

- Structured programming languages
- Design methods and principles
 - Information hiding, top-down refinement, functional decomposition
- Definition of software lifecycles
 - Waterfall, spiral,...

Week 1, Monday Lecture Slide 11

More Recent Research Results

- Process Modeling and Support
- Process Improvement
- Metrics and Empirical Studies
- Process Prescriptions

Week 1, Monday Lecture Slide 12

Process Modeling and Support

- PML (Process Modeling Languages)
- Process-centered Software Engineering Environments (PSEE)
 - Prescriptions for tools, artifacts, people, and interrelations
- Uses for process models
 - Understanding
 - Process design
 - Training and education
 - Simulation and optimization

Week 1, Monday Lecture Slide
13

Process Improvement

- Assessment of existing process
- Programs for improving software process
 - Capability Maturity Model (CMM)
 - ISO
 - Six Sigma
 - Metrics-based programs (e.g. Grady and Caswell at HP)

Week 1, Monday Lecture Slide
14

Metrics and Empirical Studies

- Quantitative measures for improving quality of software product and predictability of software process
- Definition of Metrics
- Empirical Methods for Evaluation
- Evidence of Efficacy

Week 1, Monday Lecture Slide
15

Processes

- Definition of new (prescriptive) processes
 - Personal Software Process (PSP)
 - Unified Software Development Process (OMG)

Week 1, Monday Lecture Slide
16

Criticisms

- Lack of technology transfer to industry
- Decreasing number of publications in journals and conferences
- Community stagnation

Week 1, Monday Lecture Slide
17

Future Directions

- Software processes are processes too
- The purpose and nature of PMLs/PSEEs must be rethought
- Empirical studies are a means, not an end
- Software process improvement is process improvement too

Week 1, Monday Lecture Slide
18

Software processes are processes too

- Good process research is not unique to software
 - Lots of research into processes in other disciplines
 - Arrogant not to be drawing on them

Week 1, Monday Lecture Slide
19

Purpose and Nature of PMLs/PSEEs

- PMLs need to become easier to use, more intuitive, and more accommodating to users and process enactment
- PSEEs need to become less complex and less intrusive
 - Example: configuration management tools
 - Automate tedious tasks, support social interactions

Week 1, Monday Lecture Slide
20

Empirical studies are a means, not an end

- Empirical studies must produce results that more significant (size) and have better external validity.
 - Example: Recently accepted journal paper with 50 pages of data and statistics to say that adoption of requirements engineering techniques is correlated with improved software process
 - Example: Parnas's principle of information hiding would not be published today due to lack of empirical validation

Week 1, Monday Lecture Slide
21

Software process improvement is process improvement too

- Software process is situated in other processes
 - Social processes
 - Business processes
 - Engineering and design processes
- Not special or isolated; need to draw on these for technologies and methods as well.
- Software development is a highly creative, human-centered activity, as are many other engineering and design processes in our society

Week 1, Monday Lecture Slide
22