eXtreme Programming: The Role of Inspection & Review

Prof. Dr. Dieter Rombach

Presentation

TK1

Friday 23rd November, 2001

International Conference On
Software Testing, Analysis & Review
November 19 - 23 Stockholm, Sweden

Thursday 22 November 2001

Keynote 1

Extreme Programming: The Role of Inspection & Reviews

Prof. Dr. Dieter Rombach

Dr. H. Dieter Rombach is a Full Professor in the Fachbereich Informatik (i.e., Department of Computer Science) at the Universität Kaiserslautern, Germany. He holds a chair in software engineering, is a co-director of a basic engineering research institute (SFB), and is director of the Fraunhofer Institute for Experimental Software Engineering (IESE) which aims at shortening the time needed for transferring research technologies into industrial practice. His research interests are in software methodologies, modeling and measurement of the software process and resulting products, software reuse, and distributed systems. Results are documented in more than 120 publications in international journals and conferences.

Prior to his current position, Dr. Rombach held faculty positions with the Computer Science Department and UMIACS (University of Maryland Institute for Advanced Computer Studies) at the University of Maryland, College Park, Maryland [1984-1991] and was a project leader in the SEL (Software Engineering Laboratory, a joint venture between NASA, Goddard Space Flight Center, Computer Sciences Corporation, and the University of Maryland).

He received his BS degree in mathematics from the University of Karlsruhe, Federal Republic of Germany, in 1975, his MS degrees in mathematics and computer science from the University of Karlsruhe in 1978, and his Ph.D. degree in computer science from the University of Kaiserslautern, Federal Republic of Germany, in 1984. In 1990 he received the prestigious Presidential Young Investigator Award (US$ 500,000.00) from the National Science Foundation in recognition of his research accomplishments in software engineering (especially for his NASA work). In 2000, Professor Rombach was awarded the Rhineland-Palatinate State Service Metal.

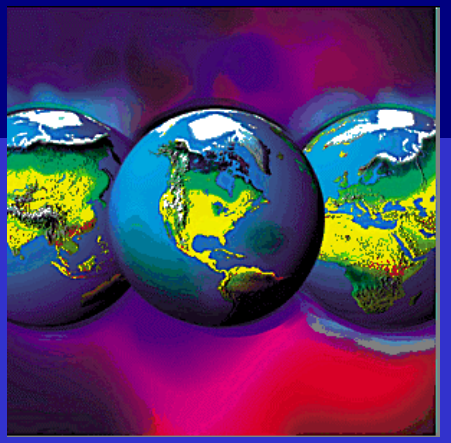
Dr. Rombach heads several research projects funded by German Government, European Union and Industry. He consults for numerous companies on issues including quality improvement, software measurement, software reuse, process modeling and software technology in general. He frequently gives industrial seminars on software quality improvement, software measurement, software reuse, and process modeling. He was Co-Guest-Editor of two Special Issues in IEEE Software, on Software Quality Assurance in September 1987 and Measurement-Based Process Improvement in July 1994, respectively, and organized the International Workshop on Experimental Software Engineering Issues in Dagstuhl, Germany, September 1992. He served as General Chair of the 18th International Conference on Software Engineering in Berlin, 1996. He is an associate editor of the Kluwer Journal "Empirical Software Engineering" and serves on the editorial boards of numerous other journals and magazines. He is a member of GI, IEEE and ACM.

EuroSTAR 2001

eXtreme Programming:

The Role of Inspection & Testing?

22nd November 2001, Sweden.



Contents:

1. Motivation
2. Product vs process focused development
3. Inspections & Testing
4. eXtreme Programming
5. Synergies
6. Risks
7. Summary
8. Future Directions

Motivation (1/2):

1. Early defect detection via inspections pays
2. Positive experiences exist from rigid (quality) process contexts
   → Cleanroom (e.g., NASA)
   → Fraunhofer IESE (e.g., Allianz)
3. Negative experiences exist from low maturity contexts
   → no process adherence
   → not lived/sustained under project pressure

Motivation (2/2):

4. eXtreme Programming (XP) process
   - → focuses on cycle time
   - → incorporates sound software engineering principles (e.g., peer reviews)
   - → lacks precise feedback & control mechanisms
5. Danger exists that
   - → cycle times gets reduced
   - → at the expense of quality (guarantee)
6. Who can benefit from XP under what circumstances?

Product vs process focused development (1/2):

1. Process-focused development (classic economy)
   - → quality (e.g., security) levels must be guaranteed
   - → automotive, medical devices domains
   - → cost of defects is enormous (e.g., liability, cost of re-call actions)
2. This requires
   - → defined processes (i.e. entry/exit criteria)
   - → measurement-based tracking
   - → continuous improvement
   - → explicit models (to repeat)

Product vs process focused development (2/2):

3. Product-focused development (new economy)
   - → time of market entry determines market share
   - → customers (at list initially) compromise on qaulity
   - → software as product, innovative services
4. This requires
   - → defined (but flexible) processes
   - → measurement-based tracking (focus?)
   - → continuous improvement (focus?)
   - → explicit models (to repeat) (focus?)
5. Most companies need to be both
   - → initially product focused (to penetrate markets)
   - → later process focused (to create revenues)

Inspections & Testing (1/3):

1. Inspections find defects
   → earlier
   → more effectively
   → more efficiently (lower cost)
   → with less distortion of system architecture and documentation
   → repeatably with tractable reading techniques

Inspections & Testing (2/3):

2. Testing
   → better suited for quality assessment
   → more effective and efficient for certain defect types
3. Knowledge about most effective/efficient defect detection time & technique could optimize cost and time
   → concentrate on 'specific' defect types during inspections/peer review
   → leave other defect types until testing

Inspections & Testing (3/3):

4. Such optimization decisions require knowledge about
   → types of defects
   → infusion vs. detection times
   → costs according to types & detection times
5. Such information can be gained from measurement-based improvement programs
6. Examples

eXtreme Programming:

1. Software development process contains sound software engineering principles
   → ….
   → peer reviews (or inspections)
   → …
2. Lack of criteria for
   → HOW should peer reviews be conducted?
   → what amount of effort pays?
3. Quality guarantee requires well-defined engineering process

Synergies:

1. Peer reviews could be
   → defined via reading techniques
   (tractable and measurable)
   → controlled in order to optimize its use
   (balancing time, cost & quality goals)
   → treated as manageable process asset to repeat results
   (knowing its effect under varying circumstances)
   → improved and sustained
2. XP could be customized to domains

- XP could be excluded from domains

Risks:

1. XP is a substitute for low-maturity software development
2. XP becomes a silver bullet
   → creates harm
   → dies for the wrong reasons
3. XP will not be used in semi-critical applications (e.g., banking)

Summary:

1. XP combines proven software engineering principles
2. XP needs to be operationalized
   for different domains
3. Peer reviews could be operationalized based on existing
   experiences from inspections & reading
4. Different forms of inspection rigor could help instantiate
   XP for different quality needs in different domains
5. This requires measurement-based defect models
6. What domains are not suitable for XP?
   → embedded critical systems?
   → none, if XP with rigorous (find all defects attitude)
   peer reviews is used?

Future Directions:

1. More empirical evidence about the cost of early vs. late defect detection
   → different defect types
   → cost of defect lag time
2. Focusing of testing based on inspection results
   → to minimize the cost of defects
3. Manageability of ‘loose’ life cycle models
   → harder than rigid models
   → experiences have been gained from early incremental developments in Cleanroom
4. Towards an XP portfolio of variants for different software domains