Brady Field

Software Engineering

**Important points**

“Software Engineering is a layered technology. Any engineering approach (including software engineering) must rest on an organizational commitment to quality.” [3]

“A process framework establishes the foundation for a complete software process by identifying a small number of framework activities that are applicable to all software projects, regardless of their size or complexity.” [4]

“Intelligent application of any software process model must recognize that adaptation (to the problem, to the project, to the people doing the work, and to the organizational culture) is essential for success.” [5]

“A process model for software engineering is chosen based on the nature of the project and application, the methods and tools to be used, and the controls and work products that are required.” [6]

“The incremental model combines elements of the waterfall model applied repetitively in an iterative fashion.” [7]

“Evolutionary models are iterative. They are characterized in a manner that enables software engineers to develop increasingly more complete versions of the software.” [8]

“The spiral model, originally proposed by Boehm, is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model.” [9]

“The spiral model is a realistic approach to the development of large-scale systems and software. Because software evolves as the process progresses, the developer and customer better understand and react to risks at each evolutionary level.” [10]

“The inception phase of the UP encompasses both customer communication and planning activities. By collaborating with the customer and end users, business requirements for the software are identified, a rough architecture for the system is proposed, and a plan for the iterative, incremental nature of the ensuing project is developed.” [11]

“The production phase of the UP coincides with the delivery and feedback activity of the generic process. During this phase, the ongoing use of the software is monitored, support for the operating environment (infrastructure) is provided, and defect reports and requests for changes are submitted and evaluated.” [12]

“Agile development can provide important benefits, but it is not applicable to all projects, all products, all people, and all situations.” [13]

“In XP, design is viewed as a transient artifact that can and should be continually modified as construction proceeds. The intent of refactoring is to control these modifications by suggesting small design changes that ‘can radically improve the design’. It should be noted, however, that effort required for refactoring can grow dramatically as the size of an application grows.” [14]

“Effective software project management focuses on the three Ps: people, problem, and process. The order is not arbitrary.” [15]

**Disagreements**

“Although not a mainstream approach, the formal methods model offers the promise of defect-free software.” [11]

While I can see the appeal of such a rigorous method, there is no such thing as defect-free software. This can be a great tool to minimize many defects in appropriate situations, but it should not be viewed as a ‘silver bullet’ solution since no such solution exists. Software will always contain defects. This is a key problem of software engineering. We are always choosing between different possibilities. These possibilities all have their own strengths and problems. We attempt to maximize the strengths and minimize the drawbacks with the resources we have. This cannot be avoided. We will never have defect-free software.

**Questions**

I have no questions. I understood everything in the article.