David Lambertson

**Software Engineering Summary**

**Important Points from Article**

* “The foundation for software engineering is the process layer.” [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]
* “Process models that stress detailed definition, identification, and application of process activities and tasks have been applied within the software engineering community for the past 30 years.” [6]
* “All prescriptive process models accommodate the generic framework activities that have been described earlier, but each applies a different emphasis to these activities and defines a workflow that invokes each framework activity (as well as software engineering actions and tasks) in a different manner.” [6]
* “Today, software work is fast-paced and subject to a never-ending stream of changes (to features, functions and information content).” [7]
* “The incremental model combines elements of the waterfall model applied repetitively in an iterative fashion. The incremental model (Figure 4) applies linear sequences in a staggered fashion as calendar time progress.” [7]
* “Evolutionary models are iterative. They are characterized in a manner that enables software engineers to developed increasingly more complete versions of software.” [8]
* “Although prototyping can be used as a stand-alone process model, it is more commonly used as a technique that can be implemented within the context of any one of the process models noted in this paper.” [8]
* “Using the spiral model, software is developed in a series of evolutionary releases.” [9]
* “It is evolutionary in nature, demanding an iterative approach to the creation of software.” [10]
* “Those localized software characteristics are modeled as components (e.g., object-oriented classes) and then constructed within the context of a system architecture.” [11]
* “The UP [Unified Process] recognizes the importance of customer communication and streamlined methods for describing the customer’s view of a system (the use case).” [11]
* “A manifesto is normally associated with an emerging political movement, one that attacks the old guard and suggests revolutionary change (hopefully for the better). In some ways, that is exactly what agile development is all about.” [12]
* “The most widely discussed agile process model is Extreme Programming (XP). XP uses an object-oriented approach as its preferred development paradigm and encompasses a set of rules and practices that occur within the context of four framework activities: planning, design, coding, and testing.” [13]
* “A central notion in XP is that design occurs both before and after coding commences. Refactoring means that design occurs continuously as the system is constructed.” [14]

**Things I Didn't Agree With**

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

I disagree with this statement because there is no such thing as ‘defect-free software’. You can comb over the software for hours and get many of the bugs, but there will always be that one which everyone misses. Humans are error prone. If we can fail, it usually happens. Due to us being prone to error, software we write will have errors in it and it is not worth the resources to find all the errors.

**Things I Did Not Understand**

I understood the whole article.