



---

Software Engineering Term Paper

**Term Paper 2003:  
My Best View of a Topic in Software Engineering**

**Review Report**

**Article Focus: Engineering Internet Software**

**Report Author**

**Corey Chong Sheng Jiat**

**Tutorial Group S8**

**Article Author**

**Scott W. Ambler**, *Ronin International*

**“Lessons in Agility from Internet-Based Development”**

**March/April 2002**

**Article Information**

IEEE Software

On page(s): 66-73

March/April 2002

Volume: 19 Issue: 2

References Cited: 17

**Key Words**

Agility, Agile Modeling, AM, Extreme Programming, XP, Dynamic System, system architecture, methodology, traditional methodology, practices, principles, internet-based, development, software engineering, internet software, system design, adaptive process.

**Synopsis**

In year 2000, internet development is on its peak. Companies widely use the net to effectively develop its systems and architecture. The author touches on efficient communication that can be produced from internet based documentation that can be accessed by every developer. Periodic on-line reviewing of requirements, planning and modeling enable the companies to react instantaneously to changes, thus having the competitive edge over competitors and generate satisfaction for end users.

**Introduction**

My article review is based on Scott W. Ambler’s article released on March/April 2002 “Lessons in Agility from Internet-Based Development”. The article presented the author’s insight on two dot-com companies that are still on business which he calls XYZ.net and PQR.com. He elaborates on the projects undertaken by the companies to re-develop their system architecture during their business startup period. His experience in the two projects provided him with the insights to come up with a new methodology that he calls “Agile Modeling methodology”.



## Software Engineering Term Paper

### Contribution to Software Engineering

Agile Modeling (AM) is a methodology used by software professionals for effective modeling and documentation of software systems. It is a 'tool' to be used in practice. To create certain model of a particular system AM does not provide hard and fast rules or detailed procedures. Instead, it lays down practices, principles and advice to let modelers know how to be effective. Basically traditional modeling methodologies are just not good enough. Thus, AM provides complementary advice to improve effectiveness towards modeling. Traditional methodologies include Extreme Programming (XP), Scrum and Dynamic System Development Method. Below lists the principles and practices of Agile Modeling:

#### Agile Modeling Technique Elaboration

<b>Core Principles</b>	<b>Elaboration</b>
• <b>Assume simplicity</b>	Do not overbuild software and not creating models that will be left unused for the present. Be confident that the system can be modified in the future as requirements evolve. Key is to be as simple as possible.
• <b>Embrace change</b>	Viewpoints do change and requirements change over time as the project moves on. What is important is that the project's environment changes proportionally as development progresses.
• <b>Incremental change</b>	Important concept for modeling is that you wouldn't need to get it right for the first time. Changes should be developed incrementally as project evolves with time.
• <b>Model with purpose</b>	It is important to identify a valid purpose in creating a model and the audience for that model. This also applies to change to an existing model.
• <b>Multiple models</b>	Having a wide range of models to define different aspects of a system will meet the needs of complex modern day software.
• <b>Quality work</b>	To increase effectiveness, quality should be a consistent factor. This will allow better understanding and enables easy updating or changes in the future.
• <b>Travel light</b>	Every artifact that is created, and then decided to keep, will need to be maintained over time. Lesser models allow agile modifications. The lesser the burden, the less problems the team will need to tackle with in the future.



Software Engineering Term Paper

Cont'd

<ul style="list-style-type: none"> <li>• <b>Core Principles</b></li> <li>• <b>Rapid feedback</b></li> <li>• <b>Software is your primary goal</b></li> <li>• <b>Enabling the next effort is your secondary goal</b></li> </ul>	<p><b>Elaboration</b></p> <p>The time between an action and the feedback on that action is critical. Understand client’s requirements, to analyze those requirements, or to develop a user interface that meets their needs, provides opportunities for rapid feedback.</p> <p>The goal of software development is to produce software that meets the needs of project stakeholders in an effective manner. Any activity that does not directly contribute to this goal should be questioned and avoided.</p> <p>Part of fulfilling the needs of your project stakeholders is to ensure that your system is robust enough so that it can be extended over time. Factors that need to be considered include whether the existing team will be involved with the next effort, the nature of the next effort itself, and the importance of the next effort to the organization.</p>
-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

**Identification of Related Work Published**

Author	Year	Some Description	Relation to Main Article
Scott W. Ambler	March April 2002	Because the scope of XP is much greater than that of AM, XP covers the full development life cycle; it is a candidate "base process" into which the techniques of AM may be tailored.  <a href="http://www.agilemodeling.com/essays/agileModelingXP.htm">http://www.agilemodeling.com/essays/agileModelingXP.htm</a>	Shows modeling are a part of XP and there is potential of AM to add value to XP based projects.
Martin Fowler	June 2002	Reasons for agile methods, focusing not so much on their weight but on their adaptive nature and their people-first orientation. A summary and references to the processes in this school and consideration of the factors.  <a href="http://www.martinfowler.com/articles/nwMethodology.html">http://www.martinfowler.com/articles/nwMethodology.html</a>	Given us an idea to the different adaptive methodologies like XP, Scrum, Open Source etc that have been mentioned in Ambler’s article.



---

## Software Engineering Term Paper

### **With Reference To Laboratory Project**

Having gone through this unique methodology for effective modeling, I find that to use this method to work for our laboratory project would be equivalent to the analogy of using a hammer to break an egg. In other words, our project size is simply just too small and the corresponding cost estimation metric calculated is only 48.6 person months. Nevertheless, I was profoundly amazed by the author's ability to come up with an effective methodology all on his own through experiences in his field of work. With regards to Agile Modeling, what we have touched on is based largely on commonsense and provides timely reminders to modelers to look out for such things so as to be effective.

I have also read with interest on an article by Martin Fowler who mentioned that "if you have more than fifty people you should use a traditional predictive process and if you change changing requirements you should use an adaptive process." Adaptive Processes mentioned here include Crystal, Extreme Programming (XP) and Scrum methodologies. Thus, with our project group's current progress following traditional methodology, Boehm's Spiral Model will be fine.

Please see our website: <http://ntuboggle.tripod.com/developmentApproach.html>

### **List of Possible Extensions to Agile Modeling and Improvement on This Article**

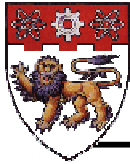
A quote "Reuse the wheel, don't reinvent it", taken from ***Lessons in Agility from Internet-Based Development*** shall begin with this requirement of the term paper. What ideas I can offer shall be with reference to my current scope of study and the limitations of my software engineering laboratory project.

A good extension would be to include descriptions on Extreme Programming (XP) which the author explicitly mentioned of AM widely adopting values of XP. Readers would be interested to know how the author derives AM from XP.

Since this article is written in the year 2002, it would be beneficial to readers to be informed of the truly effectiveness of AM if practical results of current companies after using AM are included. It would be clearer for actual software professionals having used the methodology to model their systems to describe the process, advantages and improvements that they think will help further improve AM.

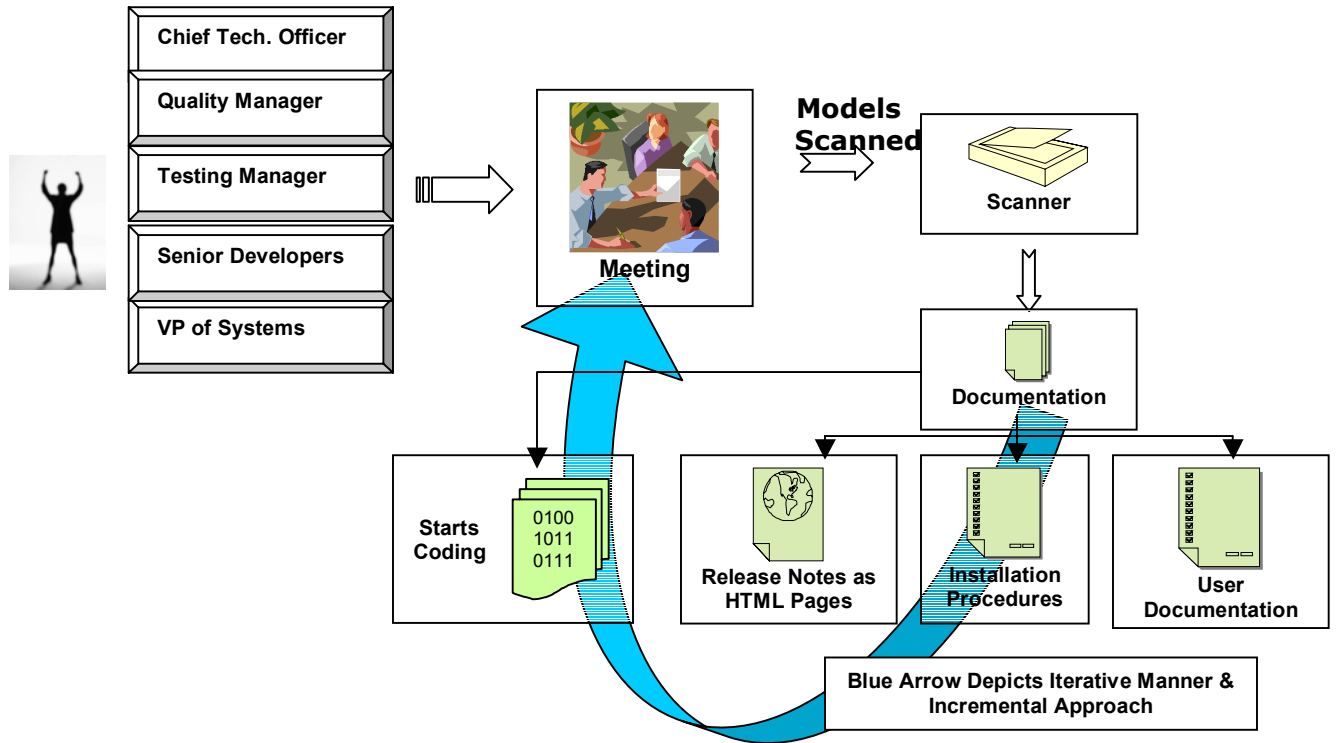
Depicting the different techniques undertaken by the two dot-com companies, I have the following extension to present the approaches in diagram form so as to help reader depict the author's perception in an easier, fast-reading form.

*Cont'd next page: Diagram depiction of approaches in article*

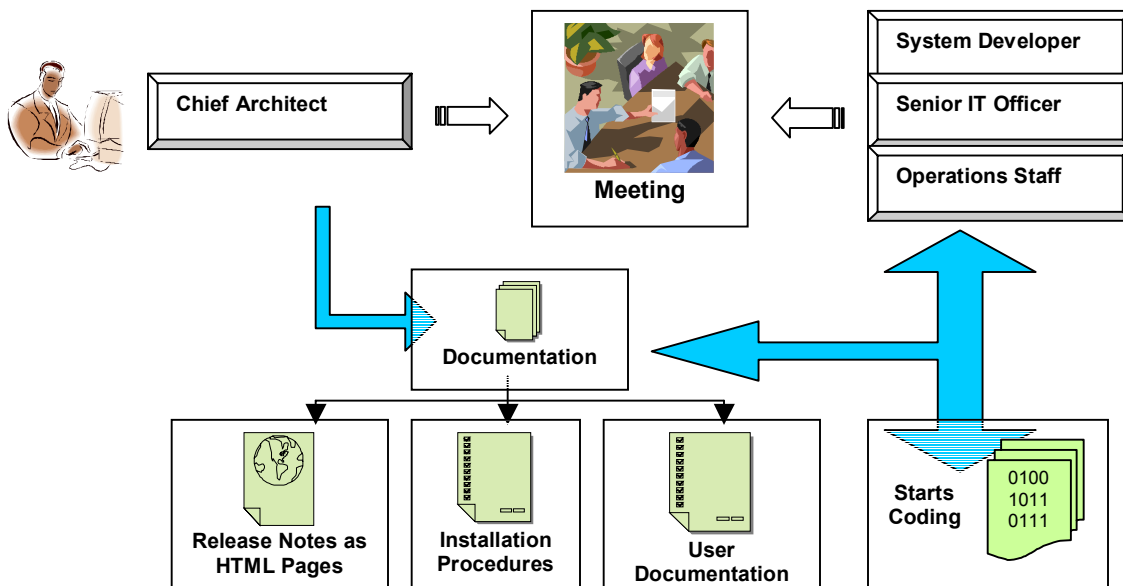


### Software Engineering Term Paper

#### XYZ.net (Communal Architecture Approach)



#### PQR.com (Common Architect Approach)





---

## Software Engineering Term Paper

### **Comments about the Paper and Relevance to Immediate Future**

This paper opens up my view on software engineering. SE appears to be nonsensical and useless documentation of methods and analysis to the typical hard-core programmer. However, after gone through the course and detailed reading of this paper, it has changed my perceptions. Every methodology appeals to the commonsense and only by documenting it will learners and practitioners alike, be able to remember it, review it, practice it and put to good use in modeling.

Computer studies and technology with regards to software engineering is still, in my view, in the infant stage. Much progress have been made over the years, still software science have much room for improvement. Extensive research work is waiting to be done, to eliminate erroneous methods and evolve primitive processes. Agile Modeling definitely represents a step towards better software engineering processes. It has provided professionals with practices and guidelines to follow, which can be further refined through experiences using it.

### **Acknowledgements**

Associate Professor Edmond C. Prakash

- For giving me this opportunity to appreciate software engineering

Ambler, S.W.

- For making a significant contribution to software engineering



---

Software Engineering Term Paper

**References**

1. <http://www.agilemodeling.com/essays/whatIsAM.htm>
2. <http://www.agilemodeling.com/essays/agileModelingRUP.htm>
3. Agile Alliance (2001a). *Manifesto for Agile Software Development*. [www.agilealliance.org](http://www.agilealliance.org)
4. Agile Alliance (2001b). *Principles: The Agile Alliance*. [www.agilealliance.org/principles.html](http://www.agilealliance.org/principles.html)
5. Ambler, S.W. (1995). *The Object Primer: Application Developer's Guide to Object Orientation*. New York: Cambridge University Press.
6. Ambler, S.W. (1997). *Building Object Applications That Work: Your Step-By-Step Handbook for Developing Robust Systems with Object Technology*. New York: Cambridge University Press. [www.ambysoft.com/buildingObjectApplications.html](http://www.ambysoft.com/buildingObjectApplications.html)
7. Ambler, S. W. (1998). *Process Patterns – Building Large-Scale Systems Using Object Technology*. New York: Cambridge University Press. [www.ambysoft.com/processPatterns.html](http://www.ambysoft.com/processPatterns.html)
8. Ambler, S. W. (1999). *More Process Patterns – Delivering Large-Scale Systems Using Object Technology*. New York: Cambridge University Press. [www.ambysoft.com/moreProcessPatterns.html](http://www.ambysoft.com/moreProcessPatterns.html)
9. Ambler, S.W. (2001a). *The Object Primer 2<sup>nd</sup> Edition: The Application Developer's Guide to Object Orientation*. New York: Cambridge University Press. [www.ambysoft.com/theObjectPrimer.html](http://www.ambysoft.com/theObjectPrimer.html)
10. Ambler, S.W. (2001b). *Enterprise Unified Process White Paper*. [www.ronin-intl.com/publications/unifiedProcess.htm](http://www.ronin-intl.com/publications/unifiedProcess.htm)
11. Ambler, S.W. (2001c). *Agile Modeling Home Page*. [www.agilemodeling.com](http://www.agilemodeling.com)
12. Ambler, S.W. (2001d). *The Design of a Robust Persistence Layer*. [www.ambysoft.com/persistenceLayer.html](http://www.ambysoft.com/persistenceLayer.html)
13. Ambler, S.W. (2001e). *Mapping Objects to a Relational Database*. [www.ambysoft.com/mappingObjects.html](http://www.ambysoft.com/mappingObjects.html)
14. Ambler, S.W. (2001f). *Agile Modeling Mailing List Instructions*. [www.agilemodeling.com/feedback.htm](http://www.agilemodeling.com/feedback.htm)
15. Ambler, S.W. (2002). *Agile Modeling: Effective Practices for XP and RUP*. New York: John Wiley & Sons. [www.ambysoft.com/agileModeling.html](http://www.ambysoft.com/agileModeling.html)
16. Ambler, S.W. & Constantine, L.L. (2000a). *The Unified Process Inception Phase*. Gilroy, CA: CMP Books. [www.ambysoft.com/inceptionPhase.html](http://www.ambysoft.com/inceptionPhase.html)
17. Ambler, S.W. & Constantine, L.L. (2000b). *The Unified Process Elaboration Phase*. Gilroy, CA: CMP Books. [www.ambysoft.com/elaborationPhase.html](http://www.ambysoft.com/elaborationPhase.html)
18. Ambler, S.W. & Constantine, L.L. (2000c). *The Unified Process Construction Phase*. Gilroy, CA: CMP Books. [www.ambysoft.com/constructionPhase.html](http://www.ambysoft.com/constructionPhase.html)
19. Ambler, S.W. & Constantine, L.L. (2002). *The Unified Process Transition and Production Phases*. Gilroy, CA: CMP Books. [www.ambysoft.com/transitionProductionPhase.html](http://www.ambysoft.com/transitionProductionPhase.html)