

Perspective

The State of Software Development

From Dark Ages to Enlightenment

Kyle Huntley
VP, GT&S, Solutions Architect

Avanade® Perspective

The State of Software Development

Table of Contents

Introduction.....	3
Technological Maturity	3
The Dark Ages	3
The Renaissance	4
The Enlightenment	5
Software Technology Maturity	6
Software Flatland.....	7
Software Today.....	8
Software Renaissance	8

Avanade® Perspective

The State of Software Development

Introduction

The expression “[Software Crisis](#)” first emerged more than 30 years ago. In the intervening years, not much has changed.

Despite new technology, improvements in [software development](#) occurred only incrementally. New practices need to be adopted that promote the evolving state of software development. Software development remains expensive because it is hard to do. No silver bullet or transformational approach exists that will make it fundamentally easier. What will help will be the ability to focus on what needs to be done and the best way to go about it. The industry needs to understand where it is in terms of technological maturity in order to choose the techniques that will help improve the state of the art.

Technological Maturity

It helps to make broad analogies to characterize technological maturity. Consider the analogy to how we construct things. To illustrate the point, envision three time periods.

- [The Dark Ages](#)
- [The Renaissance](#)
- The Enlightenment

Software as an industry falls into a broad bell curve. Much of it encompasses the Dark Ages. The upper end of the curve enters the Renaissance.

The Dark Ages

The first emergence of technology maturity occurs in the Dark Ages. Trial-and-error, ad-hoc tools and unskilled labor characterize Dark Ages’ technology. In this analogy, Dark Ages technology resembles stacking stones, bricks or wood to build a house. Unfortunately, most software still remains there.



The Renaissance

The second generation of technology maturity takes place in the Renaissance. Codified experience, clever tools, and expert crafts-workers characterize the Renaissance technology. The Renaissance technology architects arranged chunks of stone. They drew upon a wide range of experience and incremental refinement. They used simple but powerful tools. In the process, they created some truly impressive structures. While they didn’t have such tools as calculus and modern materials science, their approach equaled their scope. The beginning of the second

generation marks where good software organizations exist today.



The Enlightenment

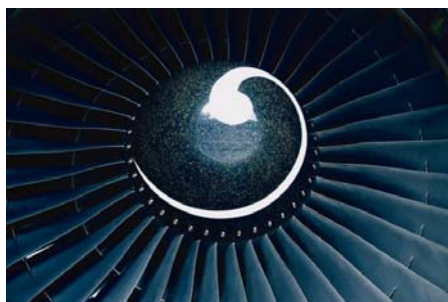
The third generation of technology maturity involves the Enlightenment. A deep systematic understanding of the problem domain, powerful tools, and standardized practices characterize Enlightenment technology. Silver bullets begin to appear with the arrival of Enlightenment technology.

A modern [skyscraper](#), or a jet engine, illustrates Enlightenment technology. The deep understanding of this domain allows the creation of buildings astonishing in scope. They stand as unique engineering marvels, built with a high degree of confidence. Being at the base of a 100-story building conveys what kind of achievement it is.

Avanade® Perspective

The State of Software Development

Enlightenment architecture transcends the limits of Renaissance architecture in the same way that chemistry surpasses alchemy. Simple repeatability replaced by genuine understanding and mastery. Software will probably never achieve this level.



Software Technology Maturity

The development of software is harder than other engineering endeavors.

It involves a different order of complexity. In most domains, the problem is less complex than the tools and techniques used to solve the problem. The tools and techniques are in a sense “complete” or “sufficient” to the problem domain.

For example, in order to build a pyramid, the job requires levels, lines, and angles as a sufficient toolset. No parts of the pyramid construction domain need more powerful conceptual tools.

Almost any physical problem can be thought of in this same way, whether it is pyramids, skyscrapers, ships or chips.

Modern engineering succeeds in this domain. The toolsets manipulate them well enough to take their place as third-generation technologies, allowing our mastery to create great works. A sense of omniscience exists with regard to the domain.

But clearly not all technologies have reached this state. Medicine, as a primary example, has reached a high level of second- generation technology. It shows great hope of genuine understanding, and the possibility of third- generation mastery. The medical tools and techniques just haven’t fully been invented yet. Bringing information technology, our most powerful tool, to the subject promises great hope for state-of-the-art advances.

But one problem domain exists that even information technology cannot fully span. That’s software, information technology itself. In general, a software problem remains unbounded. Variations in constructing a large building, for example, appear infinite. But our tools, like a CAD program, exhibit no trouble in completely expressing and mastering all possible variations.

Not so with software, however. There the problem domain is identical to the set of possible tools used to manipulate it. Information technology problems can vary infinitely. But no way exists for “getting outside” the problem. No CAD program can help because the technology and problem are intertwined.

To precisely represent a problem within the expressiveness of its own domain is to have already solved the problem, clearly a circular and unsatisfying conclusion. Imagine building a pyramid with polyhedral rocks as the only tools.

Software Flatland

An interesting perspective on the state of software engineering appears in a book published in the late 1800s. In Flatland by Edwin A. Abbott, all the people live in a two-dimensional universe. Some are triangles, some squares, hexagons, etc.

They exist in a society with artifacts, houses and roads, etc. But since everyone remains confined to a two-dimensional plane, they only see things from the side. Everything looks like a line. However, being clever, they find ways to make distinctions. They observe how the length of a line changes as it changes position, or as the subject rotates. They achieve a keen sense of distance. That allows them to detect whether a line is perpendicular to their line of sight. Through such inventive measures, and years of diligent practice, the Flatlanders act and even thrive.

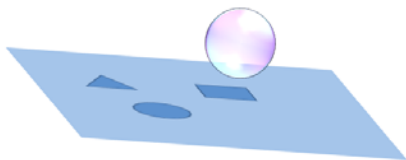
Then one day a visitor arrives in Flatland from Spaceland. This visitor, a sphere, lifts our protagonist, a square, out of his two dimensions into the space above Flatland. This new height enables him to assess his world with a new perspective. The difference between a square and a pentagon, for example, becomes obvious. This affords him many great

Avanade® Perspective

The State of Software Development

insights. Our protagonist “got outside” the problem. It allowed him to see it from a perspective that allows mastery.

To complete the analogy, trying to solve information problems with information tools leaves one stuck in Flatland. So it appears unlikely that the age of software enlightenment will occur. Software skyscrapers may be out of the question.



Software Today

A common misunderstanding exists about where software engineering sits today on the technology development scale. Some place it close to mastery.

This misunderstanding makes us optimistic that the next big thing really *is* just around the corner. It contributes to the software industry’s repeated cycle of “revolutions,” such as Structured Programming, Object Orientation, CASE tools, or Service Oriented Architecture. But these cycles aren’t revolutionary, and sometimes are just fresh expressions of well understood principles. These “revolutions” helped advance the state of the art. But in choosing the techniques needed to develop software better *now*, “revolution” is not the answer. Instead,

reliable, incremental ways to improve software economics are needed.

Software Renaissance

Is being in Software Flatland a downer? Don’t worry. The beginning of the Renaissance is an exciting time for the industry. A number of important techniques hold great potential. Rather than a “Tech X” will fix everything” approach, a set of new techniques will help individually a little, and together a lot. Failure to take advantage of these techniques, or stubbornly waiting for The Enlightenment, may never make it possible to get out of the Dark Ages of software development. Those who do adopt these techniques will see the light and realize tangible economic benefits.

Over the past nine years, Avanade’s dedicated technologists have successfully delivered over a thousand custom software applications and refined our delivery tools with a selective and highly pragmatic philosophy. From Unit Testing to Continuous Integration, from Service Oriented Architecture to Virtualization and from Aspect Oriented Programming to Domain Specific Languages, the list of tools and techniques goes on.

Avanade knows that developing software is hard and there are no silver bullets. But the industry can focus on disciplined processes and tools that bring incremental improvement. It can help us move into the Renaissance. It can help address the Software Crisis.

If you ever feel like you’re stuck in the Software Dark Ages, remember that Avanade can help you move toward the Renaissance, and maybe even aspire to Enlightenment.

Avanade® Perspective

The State of Software Development



About Avanade

Avanade provides business technology services that connect insight, innovation and expertise in Microsoft technologies to help customers realize results. Avanade's services and solutions help improve performance, productivity and sales for organizations in all industries. The company provides unsurpassed Microsoft expertise through a global network of consultants, and applies the right mix of onshore, offshore and near shore resources to deliver results faster, at lower cost and with less risk. Avanade, which is majority owned by Accenture, was founded in 2000 by Accenture and Microsoft Corporation and serves customers in more than 20 countries worldwide with more than 8,600 professionals. Additional information can be found at www.avanade.com.

©2009 Avanade Inc. All rights reserved. The Avanade name and logo are registered trademarks in the U.S. and other countries. Other brand and product names are trademarks of their respective owners.

Americas

Seattle
Phone +1 206 239 5600
America@avanade.com

Europe

London
Phone +44 0 20 7025 1000
Europe@avanade.com

Asia-Pacific

Sydney
Phone +612 9005 5900
AsiaPac@avanade.com