JOURNAL OF OBJECT TECHNOLOGY

Online at www.jot.fm. Published by ETH Zurich, Chair of Software Engineering ©**JOT, 2006**

Vol. 5, No. 4, Mai – June 2006

Breathing life into "living documents"

John D. McGregor, Clemson University and Luminary Software LLC, U.S.A.

Abstract

A multitude of sins can be hidden behind the phrase "living document." You can submit documents that are incomplete or inconsistent as long as you promise to fix it later. In this month's issue of Strategic Software Engineering, I want to talk about the strategic importance of being realistic about the state of knowledge, plans and documents in a project..

1 INTRODUCTION

When I first heard the term "living document" some years ago, I liked the concept. It signified an awareness that in an iterative, incremental development process we should be open to revisiting and revising designs, plans, and schedules. Markets become crowded, technologies emerge, and we learn. Updating strategies to reflect these new circumstances is a reasonable thing to do.

More recently I have encountered much abuse of the concept. "Don't worry, it's a living document. We can fix it later." This indicates a misunderstanding of, or disregard for, the real intent of a living document. Releasing a document that is incomplete, inconsistent, or even incorrect is risky. Any document that is released will set expectations to a degree. Future releases will not have the same impact as the original and even if the document lives on the web where changes can be made centrally, some paper copies will undoubtedly confound the change process.

Many in the United States feel that the Constitution of the United States should be considered a living document. That is, it should be interpreted in the context of today's world. On the other hand, Justice Anthony Scalia of the United States Supreme Court claims that the Constitution is not a living document. He adheres to a school of thought that the Constitution provides a fundamental definition that should not be interpreted differently in 2006 than it was 1946.

Our strategic plans have to be treated as living documents because events beyond our control occur. Supreme Court justices can ignore the changing world, we can't. But that does not mean that we have to bend to every wind of change. I want to first consider a bit of history and then propose some actions that can keep our strategic plans flexible but not chaotic.

Cite this article as follows: John D. McGregor: "Breathing life into "living documents", in *Journal of Object Technology*, vol. 5, no. 4, Mai – June 2006, pp. 17-21
http://www.jot.fm/issues/issue_2006_05/column2

2 A BIT OF HISTORY

Once upon a time we thought we did everything correctly the first time. There was no need to make changes later. There is even evidence that this was so, just look at process definitions like the old Military Standards. Hit it once and move on. No second chances, no feedback.

Of course the back channels were full of problem reports. Derived documents couldn't be mapped back to their inputs because problems were fixed in the succeeding documents but the inputs were never updated to reflect these discoveries. We simply didn't recognize these problems formally so they didn't exist – at least until the project was suddenly in deep trouble.

Maintenance was a nightmare. Since the code structure didn't conform to the architecture description, tracing errors to the associated defects was time consuming. Being certain they were fixed was even more uncertain, particularly since you couldn't count on the requirements to be complete or consistent.

Iterative, incremental processes were defined to address some of these problems. By addressing only a portion, an increment of functionality, of a product's requirements at a time these processes narrowed the focus so that a team was not trying to understand all of a product at one time. By explicitly planning iterating to make multiple passes over the development activities for an increment, these processes allowed opportunity for the learning gained in later, more detailed, activities to affect the earlier artifacts.

Boehm's spiral model and Rational's Unified Process realistically addressed the need to accommodate changes during a project. In particular, Boehm's approach explicitly addresses risk, includng the risks of not repairing previous work to reflect new realities. Both models treat previous work as "living" artifacts subject to modification.

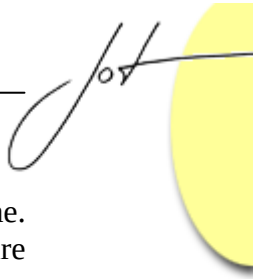
More recently, the agile approaches have focused more on incremental than iterative[1]. These process models attack a very small portion of a product's functionality at a time so that multiple passes are not necessary, although in practice completed work is often revisited and modified [Scwaber 02].

If history is any predictor, more artifacts will come alive. In 2002, the United States' CyberSecurity policy became a living document so that the policy could become "plug and play" with change made more easily. So far, at least to the best of my knowledge, it still takes a vote to change an IEEE standard!

3 EVOLUTION TRAJECTORY

Developing a product is like hitting a moving target. The larger the product and the longer the time period over which it is developed, the more opportunity there is for

[1] Note that much of the agile literature uses the term iteration for what I am calling an increment. Larman gives a brief but good discussion of this [Larman 2004, p. 10].



change and the more you have to anticipate how the target will have changed in that time. Market pressures cause requirements to change, new business goals cause the architecture to change, a new technology disrupts the normal cycle and causes previous implementation plans to change.

The iterative, incremental approach facilitates this anticipation by shortening the time and narrowing the focus. An iteration is a time slice that is a fraction of what it will take to complete the entire product. An increment focuses on only a portion of the requirements landscape for the complete product. There is less time from planning to the competed execution of the plan, leaving less time for things to change. This reduces risk to the project.

Agile methods shorten the anticipation even further. Two or three week increments are common with daily meetings to adjust the immediate trajectory [Boehm 03].

Since there are activities that begin and end an increment and an iteration, project teams develop a rhythm. “Projects in crisis have no rhythm, for they tend to be opportunistic and reactive in their work. Successful projects have a rhythm, reflected in a regular release process that tends to focus on the successive refinement of the system's architecture. This is what Microsoft calls ‘synch and stabilize,’ and it's a practice which brings results, for systems of just about any complexity.” [Booch 95]

The strategic rhythm of a project is different from the tactical rhythms. Strategic goals and decisions have more global and hopefully longer lasting implications. For example, changing the design of an algorithm because of a change in the level of performance desired can happen much more quickly, and more often, than a change from performance to security as the highest priority non-functional requirement.

In a software product line, planning for how products will be produced is a strategic action that is first conducted during the initial product line planning. Its rhythm is very slow since core assets are built and the first product built before there is any feedback to cause re-consideration. Within that strategic rhythm there is the much faster rhythm of producing the individual product production plans [Chastek 02].

4 BREATHING LIFE INTO LIVING DOCUMENTS

There is a significant difference in the effectiveness between an incomplete, or completed by guesswork, document and a living document that anticipates the normal evolution of products and domains. In this section I will present some guidelines for making living documents effective.

A living document is managed in a manner to be compatible with the evolutionary cycles of the products with which it is associated. The rhythm of the project determines when this document should be updated. Change too often and the project personnel can’t keep track of the latest information. Change too seldom and the project personnel lose confidence in the accuracy of the document.

A living document is evaluated using the C^3 criteria: correct, complete, and consistent. The Guided Inspection technique is a useful approach to addressing these criteria [McGregor 98]. If a deadline forces the release of a document before it is complete, a specific plan about when that portion will be available should be attached to the document.

A living document is accompanied by a process, usually referred to as a change management process, for handling asynchronous events that will cause the document to become obsolete. Major initiatives by competitors, new legislation, or other unanticipated events may cause the project to break out of its rhythm. The project management process should have activities to restore the rhythm of the project.

A living document is accompanied by a plan, maybe a future work section, that anticipates the evolutionary trajectory of the technologies and products that impact the document. This may be an increment plan in which specific features are assigned to the increment in which they will be introduced or a technology plan in which releases of tools from a vendor or subcontractor will enable certain project activities.

A living document lives in a controlled environment. This environment is often under the guidance of a change control board (CCB). The Board has representation from constituencies that balance the forces for and against change. The Board examines each change request to determine whether the value that is added offsets the cost. The cost includes some estimate of the time lost by document users adjusting to the change.

5 SUMMARY

The days of rigid waterfall processes and unchangeable documents that ignore reality are behind us. However, we may go to the other extreme if we are not careful. Documents that are changed on a whim confuse project personnel and often get out of synch with other rapidly changing documents. Documents that obey the rhythm of the project will facilitate an effective and efficient evolutionary trajectory. It's not a question of whether the abuse of living documents causes a project to lose its rhythm or whether once the rhythm is lost the living documents can not be coherently maintained. They are interdependent and maintenance of both the rhythm and the integrity of the documents is necessary for an effective project.

ACKNOWLEDGEMENTS

I want to thank John Hunt and Bhargavi Panjala for their comments that greatly improved this article.

REFERENCES

[Boehm 03] Barry Boehm and Richard Turner. Balancing Agility and Discipline: A Guide for the Perplexed, Addison-Wesley, 2003.

[Booch 95] Grady Booch. Object Solutions : Managing the Object-Oriented Project, Addison-Wesley, 1995.

[Chastek 02] Gary Chastek and John D. McGregor. Guidelines for Developing a Product Line Production Plan, CMU/SEI-2002-TR-006, 2002.

[Cybersecurity 02] CyberSecurity policy http://www.computerworld.com/securitytopics/security/story/0,10801,72108,00.html.

[Larman 04] Craig Larman. Agile and Iterative Development. Addison-Wesley, 2004.

[McGregor 98] John D. McGregor. The Fifty Foot Look at Analysis and Design Models, Journal of Object-Oriented Programming, 1998.

[Scwaber 02] Ken Schwaber and Mike Beedle. Agile Software Development with Scrum, Prentice Hall, 2002.

About the author

Dr. John D. McGregor is an associate professor of computer science at Clemson University and a partner in Luminary Software, a software engineering consulting firm. His research interests include software product lines and component-base software engineering. His latest book is *A Practical Guide to Testing Object-Oriented Software* (Addison-Wesley 2001). Contact him at johnmc@lumsoft.com.