

KẾT HỢP KIẾN TRÚC VÀ MÔ HÌNH AGILE VÀO PHÁT TRIỂN PHẦN MỀM CHẤT LƯỢNG CAO

INTEGRATION OF ARCHITECTURE AND AGILE MODEL INTO HIGH-QUALITY SOFTWARE DEVELOPMENT

NGUYỄN THẾ QUANG^(*) và BÙI MINH PHỤNG^(**)

TÓM TẮT: Mô hình Agile trong phát triển sản phẩm phần mềm thường ưu tiên về việc viết mã hơn là thiết kế. Nghĩa là mô hình Agile tập trung chủ yếu vào giải quyết và làm thỏa mãn yêu cầu người dùng về các yêu cầu chức năng của sản phẩm phần mềm. Nhưng trong phát triển sản phẩm phần mềm hiện nay, nếu chỉ giải quyết các yêu cầu chức năng thì chưa đủ, mà phải giải quyết cả về vấn đề chất lượng của sản phẩm phần mềm như hiệu năng, khả năng mở rộng, dễ thay đổi, tính sẵn sàng, ... Bài viết đề xuất giải pháp kết hợp thiết kế kiến trúc phần mềm và mô hình Agile để giải quyết vấn đề chất lượng của sản phẩm phần mềm trong quá trình phát triển sản phẩm phần mềm chất lượng cao.

Từ khóa: Mô hình Agile, thiết kế kiến trúc phần mềm, phương pháp phát triển phần mềm.

ABSTRACT: Agile Model for software product development focuses on writing code rather than on design. It means Agile Model focuses primarily on resolving and satisfying user requirements for functionalities of software products. With the complexities of software products today, however, a software product which meets user functional requirements is not sufficient; it is also required to meet qualities of software products such as efficiency, scalability, modifiability, availability, etc. This paper offers a solution by combining software architectural design with Agile Model to improve the quality of software products during the process of developing high quality software products.

Key words: Agile model, software architecture and design, software development models.

1. ĐẶT VẤN ĐỀ

Bất cứ một sản phẩm nào khi đưa đến người dùng sử dụng đều phải đáp ứng nhiều tiêu chí để người dùng chấp nhận và hài lòng với sản phẩm đó, trong đó hai tiêu chí bắt buộc phải có là các chức năng (Functionalities) và chất lượng (Quality) của

sản phẩm. Chức năng ở đây là sản phẩm đó có những tính năng mà người dùng sử dụng, thao tác được trên đó để thực hiện công việc nào đó. Chất lượng ở đây là sản phẩm đó có tốt hay không, dùng có bền không, có dễ sử dụng không, có đáp ứng được nhanh chóng các yêu cầu của người

^(*) ThS. Trường Đại học Văn Lang, Email: tonghunganh@vanlanguni.edu.vn

^(**) ThS. Trường Đại học Văn Lang, Email: buiminhhphung@vanlanguni.edu.vn

dùng không,... Vậy làm sao để phát triển một sản phẩm có đầy đủ chức năng và cũng có chất lượng tốt như vậy? Đối với các dự án phần mềm, để có thể tạo ra sản phẩm phần mềm có chất lượng cao, chúng ta phải có thiết kế kiến trúc cho sản phẩm phần mềm đó.

Việc thiết kế kiến trúc phần mềm cũng tương tự như thiết kế kiến trúc trong lĩnh vực xây dựng. Trong xây dựng, thiết kế kiến trúc phải được làm trước khi bắt đầu giai đoạn xây dựng. Đối với kiến trúc sư xây dựng, bản kiến trúc là nơi mà tất cả các bên liên quan như: kỹ sư xây dựng, khách hàng, người quản lý cùng nhau thảo luận để bản thiết kế đáp ứng được yêu cầu mong muốn của khách hàng và phải đảm bảo những yếu tố chất lượng như thẩm mỹ, chống được bão, chống được động đất,...

Trong ngành công nghiệp phần mềm cũng vậy, việc thiết kế kiến trúc cũng phải được tiến hành trước khi bắt đầu xây dựng sản phẩm. Thiết kế kiến trúc phải đảm bảo được tất cả các thuộc tính chất lượng (Quality Attributes) quan trọng của phần mềm phải được giải quyết triệt để, vì nếu không, khi chúng ta thực hiện đến giai đoạn phát triển sản phẩm (lập trình), xuất hiện một thuộc tính quan trọng chưa đáp ứng được yêu cầu chất lượng thì chúng ta phải làm lại giai đoạn thiết kế kiến trúc. Vấn đề này dẫn đến hao tổn chi phí, thời gian, cơ hội,... Giống như khi ta xây một ngôi nhà có năm tầng, kiến trúc sư sẽ thiết kế sao cho nền móng phải đáp ứng được cho ngôi nhà năm tầng. Nhưng khi ta đang xây dựng tới tầng thứ hai, ba hay bốn mới phát hiện ra là nền móng không đạt được tiêu chuẩn để xây nhà năm tầng thì chúng ta buộc phải

chấp nhận xây thấp hơn? Hoặc gia cố móng? Hoặc phải phá bỏ để làm móng lại? Cho dù có cách nào thì cũng tốn chi phí, cũng tốn thời gian,...

Mô hình Agile hiện nay được sử dụng rộng rãi để phát triển sản phẩm phần mềm. Mặc dù ưu điểm là có thể dễ dàng đáp ứng các thay đổi yêu cầu của khách hàng, nhưng nó có nhược điểm là không dùng để phát triển các sản phẩm phần mềm lớn [8, tr.459], hoặc sản phẩm phần mềm đòi hỏi chất lượng cao vì Agile ưu tiên viết mã cho chức năng hơn thiết kế, việc thiết kế kiến trúc trong Agile hầu như không được đề cập [5, tr.35-59]. Một số bài báo [7, tr.497-498] và sách [1] cũng đã đưa ra việc thiết kế kiến trúc trong Agile được làm nhanh vào giai đoạn đầu tiên, nhưng cũng không đưa ra hướng dẫn thiết kế kiến trúc rõ ràng mà chỉ đi vào xây dựng hay chọn một khung sườn (Framework) để phát triển sản phẩm. Nhưng việc xây dựng hay chọn một khung sườn là việc bắt đầu hay khởi điểm của thiết kế kiến trúc. Nó chỉ là việc chọn được một mẫu cho thiết kế, mà việc làm này không thỏa mãn được hết các thuộc tính chất lượng.

2. PHƯƠNG PHÁP PHÂN TÍCH VÀ THIẾT KẾ PHẦN MỀM THEO HƯỚNG KIẾN TRÚC

Sự phát triển về phân tích và thiết kế của ngành công nghiệp phần mềm đã trải qua các giai đoạn sau:

Giai đoạn phân tích và thiết kế theo cấu trúc (Structure Analysis and Design): tập trung vào phân tích và thiết kế chức năng của phần mềm. Nhưng đến thời điểm hiện nay, ngành công nghiệp phần mềm phát triển vượt bậc và đòi hỏi những phân

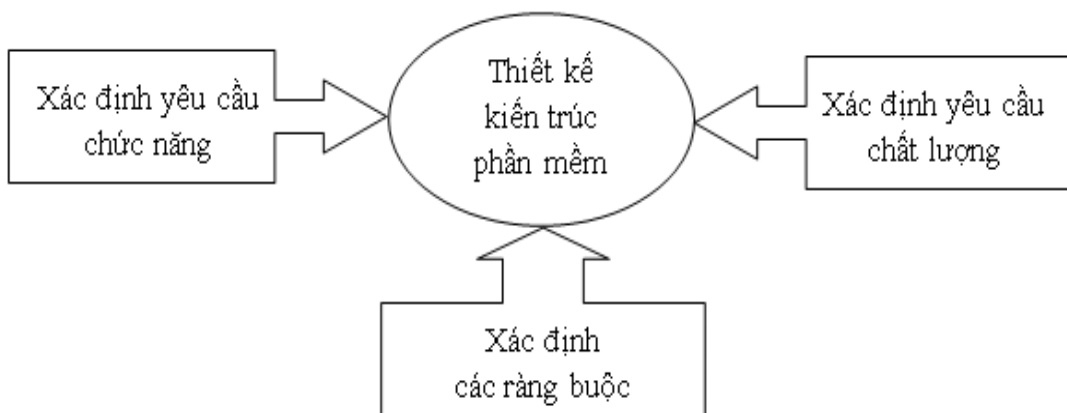
mềm phải đạt chất lượng cao, cho nên phương pháp này sẽ không còn đáp ứng được nữa.

Giai đoạn phân tích và thiết kế phần mềm theo hướng đối tượng (Object Oriented Analysis and Design): tập trung vào giải quyết chức năng lẫn chất lượng của phần mềm. Nhưng hầu như phương pháp này chỉ tập trung giải quyết chất lượng về cảnh quan tĩnh của phần mềm (khoảng 90%). Đây là phương pháp phân tích và thiết kế theo hướng đối tượng nên không áp dụng được cho các lĩnh vực mà phát triển phần mềm chất lượng cao không theo hướng đối tượng.

Giai đoạn phân tích và thiết kế theo hướng kiến trúc (Software Architecture and Design): tập trung giải quyết chất lượng và chức năng trên ba cảnh quan để đáp ứng được tất cả các yêu cầu của tất cả các bên liên quan.

Vậy, kiến trúc phần mềm được định nghĩa như sau: “*Kiến trúc phần mềm của một chương trình hay hệ thống tính toán là một cấu trúc hay các cấu trúc của hệ thống, bao gồm các phần tử phần mềm, các thuộc tính thấy được bên ngoài của những phần tử đó, và mối quan hệ giữa chúng*”.

Để thiết kế được kiến trúc phần mềm, chúng ta phải xác định được ba tiêu chí mấu chốt sau (Hình 1):



Hình 1. Ba tiêu chí cần để thiết kế kiến trúc

Xác định yêu cầu chức năng: mô tả những gì hệ thống phải làm. Ở mức kiến trúc, chúng ta chỉ xác định yêu cầu ở mức cao.

Xác định yêu cầu thuộc tính chất lượng: các đặc trưng mà hệ thống phải có bên cạnh các tính năng. Việc xác định hay để phát hiện ra thuộc tính chất lượng là một việc làm không hề đơn giản.

Xác định các ràng buộc: ràng buộc có tác động và ảnh hưởng trực tiếp đến thiết kế. Có hai loại ràng buộc:

+ *Ràng buộc về kỹ thuật:* những ràng buộc về ngôn ngữ, nền tảng, hệ quản trị, cơ sở dữ liệu. Nó là bức tường chịu lực trên không gian thiết kế.

+ *Ràng buộc về kinh doanh:* những ràng buộc về chi phí, chính sách, hoạt động của doanh nghiệp.

Sau khi xác định được ba tiêu chí trên, chúng ta tiến hành thiết kế kiến trúc cho phần mềm. Bản thiết kế phải đảm bảo được đầy đủ ba cảnh quan (mô hình thiết kế):

Cảnh quan động: *processes, threads, events, dataflows, ...*

Cảnh quan tĩnh: *classes, module, library, use, ...*

Cảnh quan vật lý: *computers, networks, routers, ...*

Việc thiết kế kiến trúc phải qua các bước sau:

Bước 1: Tạo sơ đồ ngữ cảnh.

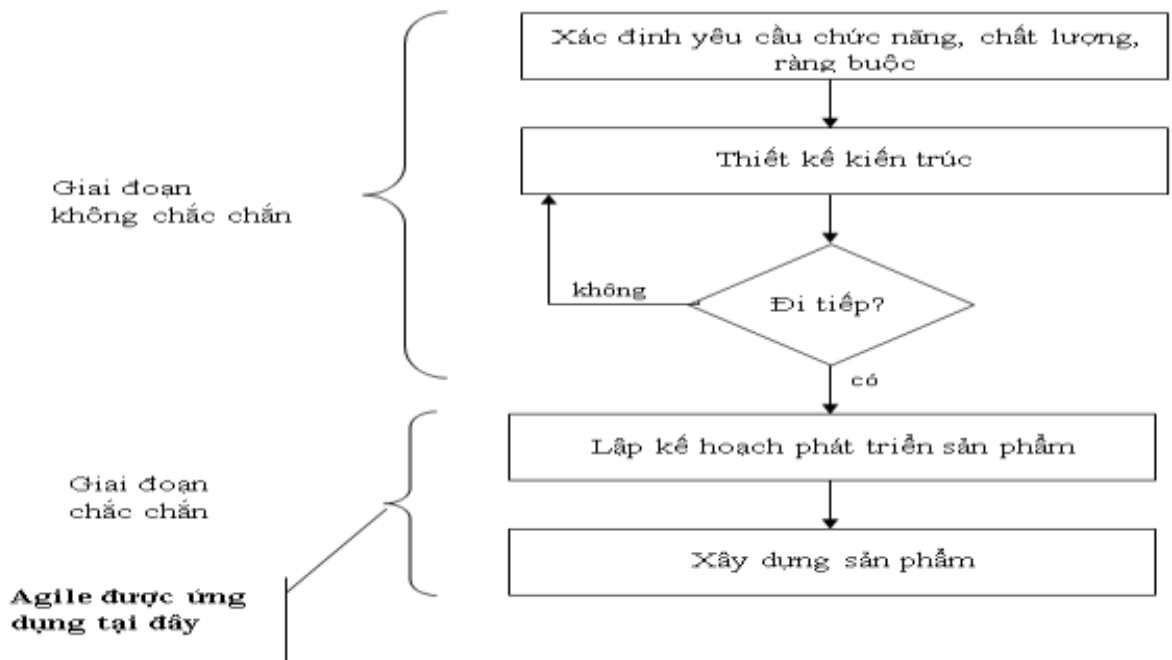
Bước 2: Chọn một cảnh quan bất kỳ và tiến hành phân rã.

Bước 3: Nếu chưa thỏa hết các thuộc tính chất lượng, chuyển sang cảnh quan khác và tiếp tục phân rã.

Bước 4: Lập lại khi cần thiết.

3. KẾT HỢP KIẾN TRÚC VÀ MÔ HÌNH AGILE VÀO PHÁT TRIỂN PHẦN MỀM CHẤT LƯỢNG CAO

Theo Anthony Latanze [2], việc phát triển phần mềm chất lượng cao sẽ trải qua hai giai đoạn: giai đoạn không chắc chắn và giai đoạn chắc chắn (Hình 2).



Hình 2. Mô tả giai đoạn sử dụng Agile

Giai đoạn không chắc chắn: là giai đoạn kiến trúc chưa hoàn thành.

Giai đoạn chắc chắn: là giai đoạn đã thiết kế xong kiến trúc cho phần mềm.

Do mô hình Agile ưu tiên viết mã cho chức năng hơn thiết kế nên chúng ta sẽ áp dụng mô hình Agile ở giai đoạn chắc chắn để phát triển sản phẩm phần mềm yêu cầu chất lượng cao.

4. KẾT LUẬN

Với những phần mềm ngày càng phức tạp về nghiệp vụ, rất khó để phát triển sản phẩm theo mô hình truyền thống, nhưng mô hình Agile cũng có nhược điểm là tập trung vào yêu cầu chức năng của sản phẩm là chính, rất khó để phân tích và đáp ứng các yêu cầu về thuộc tính chất lượng. Do đó, việc đưa thêm giai đoạn thiết kế kiến

trúc phần mềm vào trước khi áp dụng mô hình Agile trong quá trình phát triển các dự án phần mềm sẽ giúp cho việc xây dựng phần mềm có được bộ khung chắc chắn, đảm bảo các thuộc tính chất lượng.

Tuy nhiên, chúng tôi chỉ đề xuất về mô hình là đưa phần kiến trúc vào trước giai

đoạn thực hiện theo Agile mà chưa đưa ra các tiêu chí cụ thể, cũng như những đánh giá cụ thể từng bước trong mô hình này. Chúng tôi cho rằng đây là công việc nên được nghiên cứu thêm và đưa ra giải pháp cụ thể trong thời gian tới.

TÀI LIỆU THAM KHẢO

1. Coplien, James and Bjørnvig, Gertrud (2010), *Lean Architecture: For Agile Software Development*, John Wiley & Sons.
2. Lattanze, Anthony J (2008), *Architecting Software Intensive Systems: A Practitioners Guide*, CRC Press.
3. Len, Bass, Paul, Clements, and Rick, Kazman (2003), *Software architecture in practice*, Boston, Massachusetts Addison.
4. Shaw, Mary and Garlan, David (1996), *Software Architecture: Perspectives on an Emerging Discipline*, Vol. 1, Prentice Hall Englewood Cliffs.
5. Stober, Thomas and Hansmann, Uwe (2010), "Overview of Agile Software Development", *Agile Software Development*, Springer.
6. Clements, Paul, Kazman, Rick, and Klein, Mark (2002), Evaluating software architectures: methods and case studies, *Publié par Addison-Wesley Professional*.
7. Kruchten, Philippe (2010), Software Architecture and Agile Software Development: A Clash of Two Cultures?, *2010 ACM/IEEE 32nd International Conference on Software Engineering*, IEEE.
8. Mohammad, Adel Hamdan and Alwada'n, Tariq (2013), *Agile Software Methodologies: Strength and Weakness*, *International Journal of Engineering Science and Technology*. 5(3).

Ngày nhận bài: 07/11/2016. Ngày biên tập xong: 07/3/2017. Duyệt đăng: 21/3/2017