



4. Mở rộng quy mô phương pháp Agile

- ❖ Các phương pháp Agile đã được chứng minh là thành công đối với các dự án vừa và nhỏ, có thể được phát triển bởi một nhóm nhỏ cùng làm việc một chỗ.
- ❖ Đôi khi người ta lập luận rằng sự thành công của những phương pháp này là do thông tin liên lạc được cải thiện, vì mọi người làm việc cùng nhau.
- ❖ Mở rộng quy mô các phương pháp Agile liên quan đến việc thay đổi các phương pháp này để đối phó với các dự án lớn hơn, dài hơn, nơi có nhiều nhóm phát triển, có thể làm việc ở các vị trí khác nhau.



4.1 Scaling out và scaling up

- ❖ ‘Scaling up’ liên quan đến việc sử dụng các phương pháp Agile để phát triển các hệ thống phần mềm lớn mà một nhóm không thể phát triển được.
- ❖ ‘Scaling out’ liên quan đến cách các phương pháp Agile được giới thiệu trong một tổ chức lớn với nhiều năm kinh nghiệm phát triển phần mềm.
- ❖ Khi mở rộng quy mô phương pháp Agile, điều quan trọng là phải duy trì các nguyên tắc cơ bản:
 - Lập kế hoạch linh hoạt (Flexible planning),
 - Phát hành hệ thống thường xuyên (frequent system releases),
 - Tích hợp liên tục (continuous integration),
 - Phát triển hướng thử nghiệm (test-driven development) và
 - Giao tiếp nhóm tốt (good team communications).



4.2 Các vấn đề thực tế với phương pháp Agile

- ❖ Tính không chính thức của phát triển Agile không tương thích với cách tiếp cận pháp lý để xác định hợp đồng thường được sử dụng trong các công ty lớn.
- ❖ Các phương pháp Agile thích hợp để phát triển phần mềm mới hơn là bảo trì phần mềm. Tuy nhiên, phần lớn chi phí phần mềm ở các công ty lớn đến từ việc duy trì hệ thống phần mềm hiện có của họ.
- ❖ Các phương pháp Agile được thiết kế cho các nhóm nhỏ cùng vị trí địa lý nhưng hiện nay nhiều việc phát triển phần mềm liên quan đến các nhóm phân tán trên toàn thế giới.



4.3 Các vấn đề hợp đồng

- ❖ Hầu hết các hợp đồng phần mềm cho các hệ thống tùy chỉnh đều dựa trên một đặc tả, quy định này đặt ra những gì nhà phát triển hệ thống phải triển khai cho khách hàng của hệ thống.
- ❖ Tuy nhiên, điều này loại trừ các đặc điểm kỹ thuật và phát triển đan xen như là tiêu chuẩn trong phát triển Agile.
- ❖ Cần có hợp đồng trả tiền cho thời gian của nhà phát triển thay vì chức năng.
 - Tuy nhiên, đây được coi là rủi ro cao của nhiều bộ phận pháp lý vì những gì phải được chuyển giao không thể được đảm bảo.



4.4 Agile với Bảo trì phần mềm

- ❖ Hầu hết các tổ chức chi tiêu nhiều hơn vào việc duy trì phần mềm hiện có so với việc phát triển phần mềm mới. Vì vậy, nếu các phương pháp Agile thành công, chúng phải hỗ trợ bảo trì tương tự như quá trình đã phát triển.
- ❖ Hai vấn đề chính:
 - Các hệ thống được phát triển bằng cách sử dụng phương pháp tiếp cận Agile có thể bảo trì được không?
 - Có được nhấn mạnh trong quá trình phát triển là giảm thiểu tài liệu chính thức không?
 - Các phương pháp Agile có thể được sử dụng hiệu quả để phát triển một hệ thống theo yêu cầu thay đổi của khách hàng không?
- ❖ Các vấn đề có thể phát sinh nếu không thể duy trì nhóm phát triển ban đầu.



4.4.1 Bảo trì với Agile

- ❖ Các vấn đề chính là:
 - Thiếu tài liệu sản phẩm.
 - Giữ khách hàng tham gia vào quá trình phát triển.
 - Duy trì tính liên tục của nhóm phát triển.
- ❖ Phát triển Agile dựa vào việc nhóm phát triển biết và hiểu những gì phải làm.
- ❖ Đối với các hệ thống lâu đời, đây là một vấn đề thực sự vì các nhà phát triển ban đầu sẽ không phải lúc nào cũng làm việc ở đó.



4.5 Agile | PD

❖ Quyết định trọng số dựa trên:

- Quan trọng là phải có một đặc tả kỹ thuật và thiết kế rất chi tiết trước khi chuyển sang thực hiện?
 - > Vậy bạn cần sử dụng PD.
- Chiến lược phân phối gia tăng (nơi bạn cung cấp phần mềm cho khách hàng và nhận được phản hồi nhanh từ họ) có thực tế không?
 - > Nếu có, xem xét sử dụng các phương pháp Agile.
- Hệ thống đang phát triển lớn đến mức nào?
 - > Phương pháp Agile có hiệu quả nhất khi hệ thống có thể được phát triển với một nhóm nhỏ đồng địa điểm, có thể giao tiếp không chính thức.
 - > Agile có thể không thực hiện được với các hệ thống lớn, đòi hỏi các nhóm phát triển lớn, do đó nên xem xét sử dụng PD.

4.6 Lý thuyết | Thực tế với Agile (a)

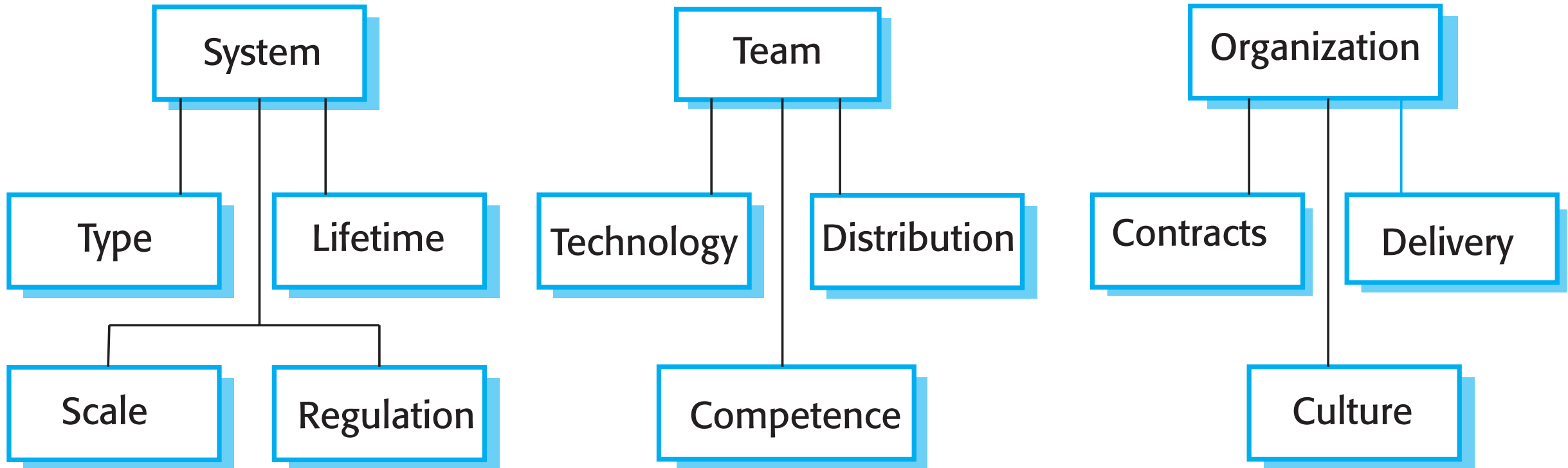
Nguyên tắc	Thực tế
Sự quan tâm của khách hàng (Customer involvement)	<p>Điều này phụ thuộc vào việc có một khách hàng sẵn sàng và có thể dành thời gian cho nhóm phát triển và người có thể đại diện cho tất cả các bên liên quan của hệ thống. Thông thường, các đại diện của khách hàng có yêu cầu khác về thời gian của họ và không thể đóng vai trò đầy đủ trong việc phát triển phần mềm.</p> <p>Khi có các bên liên quan bên ngoài, chẳng hạn như các cơ quan quản lý, rất khó để trình bày quan điểm của họ với nhóm Agile.</p>
Nắm lấy thay đổi (Embrace change)	<p>Độ ưu tiên của các thay đổi có thể cực kỳ có khấn, đặc biệt là trong hệ thống có nhiều bên liên quan. Thông thường, mỗi bên liên quan đưa ra những ưu tiên khác nhau đối với những thay đổi khác nhau.</p>
Phân phối gia tăng (Incremental delivery)	<p>Việc lặp lại nhanh chóng và lập kế hoạch ngắn hạn để phát triển không phải lúc nào cũng phù hợp với chu kỳ lập kế hoạch dài hạn của kế hoạch kinh doanh và tiếp thị. Các nhà quản lý tiếp thị có thể cần biết những đặc điểm của sản phẩm trước vài tháng để chuẩn bị một chiến dịch tiếp thị hiệu quả.</p>

4.6 Lý thuyết | Thực tế với Agile (b)

Nguyên tắc	Thực tế
Duy trì sự đơn giản (Maintain simplicity)	Dưới áp lực từ lịch trình phân phối, các thành viên trong nhóm có thể không có thời gian thực hiện đơn giản hóa hệ thống mong muốn.
Con người quan trọng hơn quy trình (People not process)	Các thành viên riêng lẻ trong nhóm có thể không có tính cách phù hợp để tham gia mạnh mẽ, và do đó có thể không tương tác tốt với các thành viên khác trong nhóm.



4.7 Các vấn đề Agile | PD





4.7.1 Vấn đề hệ thống

- ❖ Hệ thống đang được phát triển lớn đến mức nào?
 - Phương pháp Agile hiệu quả nhất khi một nhóm tương đối nhỏ có thể giao tiếp không chính thức.
- ❖ Loại hệ thống nào đang được phát triển?
 - Các hệ thống yêu cầu nhiều phân tích trước khi thực hiện cần một thiết kế khá chi tiết để thực hiện phân tích này.
- ❖ Thời gian tồn tại của hệ thống dự kiến là bao lâu?
 - Các hệ thống lâu dài yêu cầu tài liệu để truyền đạt ý định của các nhà phát triển hệ thống cho nhóm hỗ trợ sau này.
- ❖ Hệ thống có chịu sự điều tiết bên ngoài không?
 - Nếu một hệ thống được quy định, bạn có thể sẽ được yêu cầu xuất trình tài liệu chi tiết như một phần của trường hợp an toàn hệ thống.



4.7.2 Con người và đội nhóm

- ❖ Các nhà thiết kế và lập trình viên trong nhóm giỏi như thế nào?
 - Đôi khi người ta tranh luận rằng các phương pháp Agile đòi hỏi mức độ kỹ năng cao hơn so với các phương pháp tiếp cận dựa trên kế hoạch.
- ❖ Nhóm phát triển được tổ chức như thế nào?
 - Tài liệu thiết kế có thể được yêu cầu nếu nhóm làm việc phân tán.
- ❖ Có những công cụ hỗ trợ nào?
 - IDE hỗ trợ trực quan hóa và phân tích chương trình là cần thiết nếu tài liệu thiết kế không có sẵn.



4.7.3 Vấn đề tổ chức

- ❖ Các tổ chức kỹ thuật truyền thống có văn hóa phát triển dựa trên kế hoạch, vì đây là tiêu chuẩn trong kỹ thuật.
- ❖ Có phải thực hiện tổ chức tiêu chuẩn để phát triển một đặc tả hệ thống chi tiết hay không?
- ❖ Đại diện khách hàng có sẵn sàng cung cấp phản hồi về các bước tăng thêm của hệ thống không?
- ❖ Phát triển Agile không chính thức có thể phù hợp với văn hóa tổ chức của tài liệu chi tiết không?



4.8 Agile cho các hệ thống lớn

- ❖ Các hệ thống lớn thường là tập hợp của các hệ thống giao tiếp riêng biệt, trong đó các nhóm riêng biệt phát triển từng hệ thống.
 - Thông thường, các nhóm này làm việc ở những nơi khác nhau, đôi khi ở những múi giờ khác nhau.
- ❖ Hệ thống lớn là hệ thống ‘brownfield’, tức là chúng bao gồm và tương tác với một số hệ thống hiện có.
 - Nhiều yêu cầu hệ thống liên quan đến sự tương tác này do đó không thực sự cho phép tạo ra sự linh hoạt và phát triển gia tăng.
- ❖ Trong trường hợp một số hệ thống được tích hợp để ra một hệ thống mới, một phần đáng kể của quá trình phát triển liên quan đến cấu hình hệ thống hơn là phát triển mã gốc.

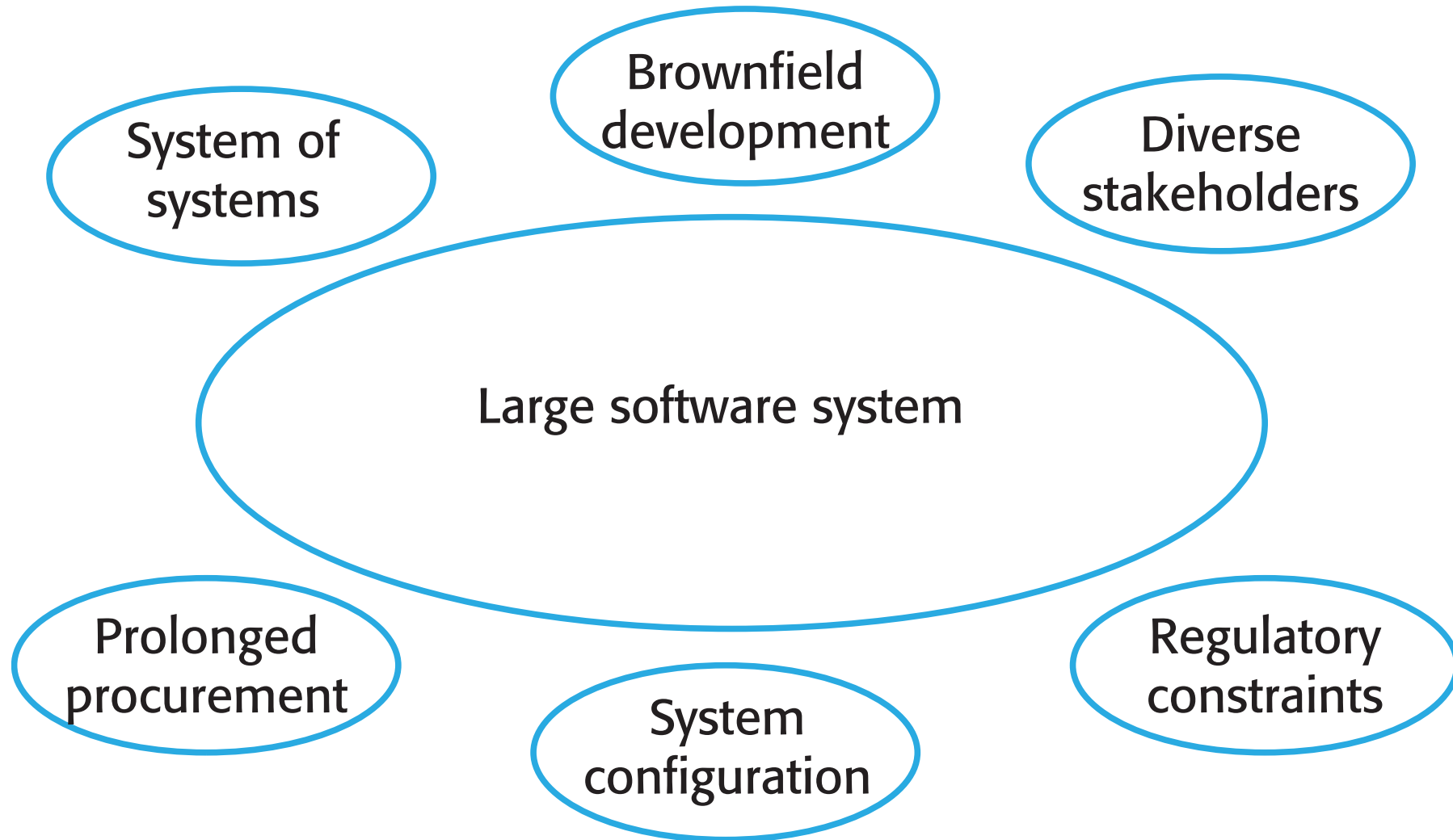


4.8.1 Vấn đề khi phát triển hệ thống lớn

- ❖ Các hệ thống lớn và quá trình phát triển của chúng thường bị hạn chế bởi các quy tắc và luật lệ bên ngoài, hạn chế cách thức mà chúng có thể được phát triển.
- ❖ Các hệ thống lớn có thời gian mua sắm và phát triển lâu dài.
 - Rất khó để duy trì các nhóm gắn kết, những người hiểu biết về hệ thống trong khoảng thời gian đó, tất yếu là mọi người chuyển sang các công việc và dự án khác.
- ❖ Các hệ thống lớn thường có nhiều bên liên quan.
 - Thực tế là không thể có sự tham gia của tất cả các bên liên quan khác nhau này vào quá trình phát triển.

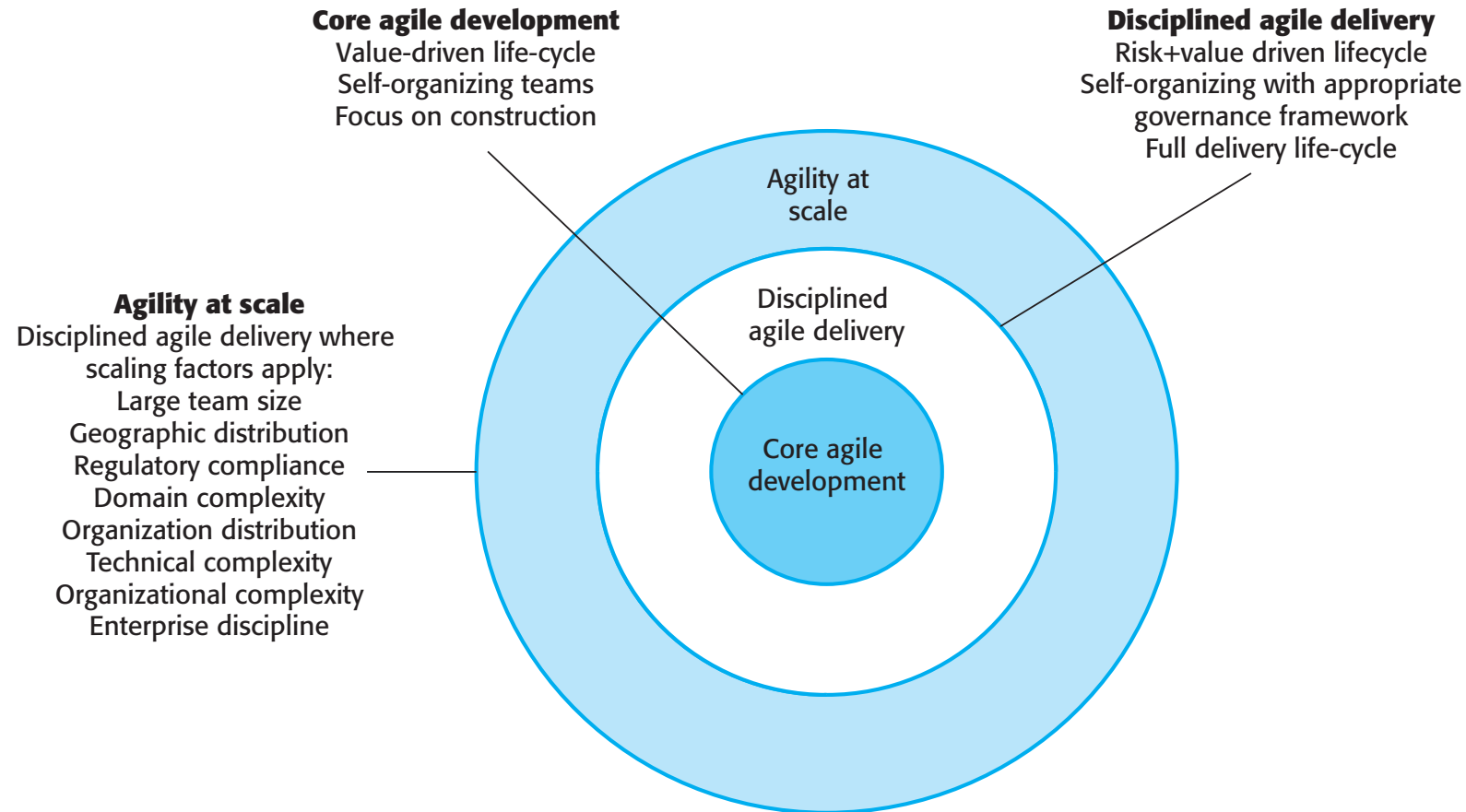


4.8.2 Các yếu tố trong hệ thống lớn





4.8.3 IBM Agile





4.8.4 Nhận định

- ❖ Cách tiếp cận hoàn toàn gia tăng đối với kỹ thuật yêu cầu là điều không thể.
- ❖ Không thể có một chủ sở hữu sản phẩm hoặc đại diện khách hàng duy nhất.
- ❖ Để phát triển hệ thống lớn, không thể chỉ tập trung vào mã hệ thống.
- ❖ Các cơ chế giao tiếp giữa các nhóm phải được thiết kế và sử dụng.
- ❖ Tích hợp liên tục trên thực tế là không thể.
 - Tuy nhiên, điều cần thiết là phải duy trì việc xây dựng hệ thống thường xuyên và phát hành hành hệ thống thường xuyên.



4.8.5 Đa nhóm Scrum

- ❖ Nhân rộng vai trò - Role replication
 - Mỗi nhóm có Product Owner cho phần công việc của họ và ScrumMaster.
- ❖ Kiến trúc sư sản phẩm - Product architects
 - Mỗi nhóm chọn một kiến trúc sư sản phẩm và những kiến trúc sư này cộng tác để thiết kế và phát triển kiến trúc hệ thống tổng thể.
- ❖ Liên kết phát hành - Release alignment
 - Ngày phát hành sản phẩm của mỗi nhóm được căn chỉnh (liên kết) để tạo ra một hệ thống hoàn chỉnh và có thể hoạt động được.
- ❖ Scrum của Scrums
 - Có một Scrum của nhiều Scrum hàng ngày, nơi đại diện của mỗi nhóm gặp nhau để thảo luận về tiến độ và lập kế hoạch công việc cần thực hiện.



4.8.6 Đưa Agile vào tổ chức

- ❖ Người quản lý dự án không có kinh nghiệm với các phương pháp Agile có thể miễn cưỡng chấp nhận rủi ro khi tiếp cận cách mới này.
- ❖ Tất cả các tổ chức lớn thường có các thủ tục và tiêu chuẩn chất lượng mà tất cả các dự án phải tuân theo và do tính chất quan liêu của chúng, những tiêu chuẩn này có thể không tương thích với các phương pháp Agile.
- ❖ Các phương pháp Agile dường như hoạt động tốt nhất khi các thành viên trong nhóm có trình độ kỹ năng tương đối cao.
 - Tuy nhiên, trong các tổ chức lớn, số lượng khả năng và kỹ năng quá nhiều.
- ❖ Có thể có sự phản đối về văn hóa đối với các phương pháp Agile, đặc biệt là trong những tổ chức có lịch sử lâu đời sử dụng các quy trình kỹ thuật hệ thống thông thường.