

## Contents

<b>B</b>	<b>2</b>
backup programmer 111 lập trình viên dự phòng.	2
<b>Tại sao cứ vị trí backup programmer lại khó tuyển ?</b>	2
Beyond Chief Programmer and Democratic Teams 113 (nhóm code có sếp kiểu mới)	2
<b>Ưu, nhược điểm của nhóm code có sếp kiểu mới (nhóm code kết hợp) là gì ?</b>	3
Brooks's Law (Luật Brooks's) 108:	3
<b>Tại sao luật Brooks's lại đúng ?</b>	3
<b>C</b>	<b>4</b>
chief programmer 111: Lập trình viên trưởng	4
chief programmer team 110 ( đội ngũ lập trình viên trưởng )	4
<b>Ưu, nhược điểm của nhóm code có sếp kiểu cũ-kiểu cổ điển là gì ? (chief programmer team).?</b>	4
<b>D</b>	<b>5</b>
democratic team 109 (nhóm dân chủ)	5
<b>Ưu, nhược điểm của nhóm code democratic là gì (nhóm code dân chủ )</b>	5
<b>E</b>	<b>5</b>
<b>egoless programming (lập trình bình đẳng) 109 *</b>	5
<b>H</b>	<b>6</b>
hierarchy (thứ bậc) 111	6
KPA - key process area (lĩnh vực tiến trình) 119	6
<b>L</b>	<b>7</b>
librarian 112 (thủ thư)	7
<b>P</b>	<b>7</b>
pair programming 118 (Lập trình cặp)	7
programmer 112 (lập trình viên)	7
programming secretary 112	7
<b>Tại sao vị trí secretary programmer cũng khó tuyển ?</b>	7
<b>S</b>	<b>8</b>
specialization 111	8
superprogrammer 113	8
<b>T</b>	<b>8</b>

task 118 (Nhiệm vụ)	8
team 107 (Nhóm đội)	8
team leader 114	8
<b>Team leader khác gì với backup programmer ?</b>	8
Team leader khác gì với secretary programming ?	8
team manager (người quản lý đội nhóm) 114:	9
<b>N</b>	<b>9</b>
<b>Nhóm code có sếp kiểu mới (nhóm code kết hợp) thì áp dụng luật Miller như thế nào ?</b>	9
Nhóm code có sếp kiểu mới (nhóm code kết hợp) thì phát huy được ưu điểm gì của nhóm code dân chủ ?	9

## B

### **backup programmer 111 lập trình viên dự phòng.**

Là người sẵn sàng thay thế lập trình viên trưởng khi lập trình viên trưởng là con người và không thể làm việc, có đầy đủ năng lực và các yếu tố của lập trình viên trưởng. Ngoài ra, để lập trình viên trưởng tập trung thiết kế kiến trúc, lập trình viên dự phòng lập kế hoạch trường hợp kiểm thử hộp đen (là hàng tỷ các trường hợp thử nghiệm) và các nhiệm vụ khác độc lập với thiết kế.

### **Tại sao cú vị trí backup programmer lại khó tuyển ?**

Vì lập trình viên dự phòng được kỳ vọng sẽ giỏi như lập trình viên trưởng nhưng phải ngồi ở ghế sau và mức lương thấp hơn trong khi phải chờ đợi có cơ hội được thể hiện khi lập trình viên trưởng đang hoạt động tốt. Rất ít lập trình viên hàng đầu hoặc nhà quản lý hàng đầu chấp nhận một vai trò như vậy

### **Beyond Chief Programmer and Democratic Teams 113 (nhóm code có sếp kiểu mới)**

Nhóm code có sếp kiểu mới (nhóm code kết hợp) là sự kết hợp các ưu điểm của nhóm code bình đẳng về tinh thần phát hiện và sửa lỗi cao và nhóm code có sếp kiểu cũ (nhóm kiểu cổ điển) về quản lý và giao tiếp tốt.

Trong nhóm code có sếp kiểu mới, để giảm bớt trách nhiệm của sếp, thì trong đó sếp chỉ quản lý các vấn đề phi kỹ thuật như là thu nhập, bình đẳng, năng lực của các thành viên, sếp sẽ tạo ra team leader để quản lý kỹ thuật: review toàn bộ code và hỗ trợ kỹ thuật cho các thành viên và sếp không review code nhưng khi họp có thể tham gia để hỗ trợ kỹ thuật cho các thành viên

Tính khả thi của nhóm code có sếp kiểu mới là:

- Tìm một team leader dễ dàng hơn nhiều một sếp
- Các thành viên chỉ bị quản lý bởi 1 sếp duy nhất
- Sếp chỉ cần kỹ năng cao về quản lý, team chỉ cần kỹ năng cao về code → dễ tìm hơn

**Ưu, nhược điểm của nhóm code có sếp kiểu mới (nhóm code kết hợp) là gì ?**

**Ưu điểm:**

- Có các ưu điểm của nhóm code bình đẳng về tính thần phát hiện và sửa lỗi cao cũng như ưu điểm của nhóm code có sếp kiểu cũ về quản lý và giao tiếp tốt
- Mỗi nhân viên chịu trách nhiệm chính xác một quản lý, trách nhiệm mỗi người là rõ ràng
- Dễ tìm và tuyển được teamleader hơn là chief programmer
- Không bị lạm phát lãnh đạo như nhóm kiểu cũ
- Lập trình viên trưởng được thay thế bằng hai cá nhân: một trưởng nhóm phụ trách các khía cạnh kỹ thuật của các hoạt động của nhóm và một người quản lý nhóm chịu trách nhiệm về tất cả các quyết định của người quản lý phi kỹ thuật. -> dễ dàng tìm kiếm nhân lực, công việc được phân định rõ ràng

**Nhược điểm:**

- Giữa 2 nhóm trưởng không đồng nhất về các vấn đề, dễ xảy ra mâu thuẫn
- Khi không phân chia việc rõ ràng, 2 sếp lãnh đạo có thể không quản lý một vấn đề do nghĩ là việc của sếp khác, hoặc 2 sếp cùng tranh nhau quản lý 1 vấn đề
- Phụ thuộc khá nhiều vào lập trình viên trưởng vì là người review toàn bộ code và hỗ trợ kỹ thuật cho các thành viên
- Người quản lý nhóm phải tham gia vào nhóm thường xuyên các cuộc họp để đánh giá các kỹ năng kỹ thuật của thành viên trong nhóm

**Brooks's Law (Luật Brooks's) 108:**

Định luật Brooks: “Bổ sung nhân lực cho một dự án phần mềm muộn khiến nó trở nên muộn hơn.” Được đặt theo tên của Fred Brooks. Theo Brooks, có một người gia tăng, khi được thêm vào một dự án, làm cho nó mất nhiều thời gian hơn, chứ không phải ít thời gian hơn. Điều này tương tự như luật chung về lợi nhuận giảm dần (law of diminishing returns) trong kinh tế.

Cùng với đó luận thuyết cơ bản của định luật này là

Cần thời gian để quen với project

Công sức dành cho việc giao tiếp sẽ tăng

**Tại sao luật Brooks's lại đúng ?**

**Để chứng minh:** Ta xét một ví dụ:

Đội phát triển có 3 người → có 3 kênh giao tiếp.

Nhưng dự án sắp đến hạn mà còn quá nhiều việc

Giải pháp trực quan: tuyển thêm 1 người → cần 6 kênh giao tiếp!

-> 3 người cũ sẽ phải diễn giải cho người mới:

Các việc đã hoàn thành

Các việc chưa hoàn thành

Cách hoàn thiện các việc còn dang dở

-> sẽ không những giải quyết được vấn đề bị trễ mà thậm chí còn mất thêm thời gian

-> Bởi vậy luật Brooks's đã nói đúng.

Đúng vì, theo luật Brooks phát biểu Bổ sung nhân lực cho một dự án phần mềm muộn khiến nó trở nên muộn hơn, ta có thể hiểu thêm như sau :

- Phải mất một thời gian để những người được thêm vào một dự án trở nên có năng suất và những người lao động mới trong dự án trước tiên phải được đào tạo về công việc đã diễn ra; đào tạo này đòi hỏi phải chuyển hướng các nguồn lực đang làm việc cho dự án, tạm thời làm giảm năng suất
- Chi phí truyền thông giao tiếp tăng khi số người tăng. Số lượng kênh truyền thông tăng nhanh theo số lượng người, vì vậy khi có thêm nhiều người, họ dành nhiều thời gian hơn để cố gắng tìm hiểu những gì mọi người khác đang làm.
- Thêm nhiều người hơn vào một nhiệm vụ có khả năng phân chia cao,. Tuy nhiên, các hoạt động khác bao gồm nhiều chuyên ngành trong các dự án phần mềm mà khó phân chia.

## C

### chief programmer 111: Lập trình viên trưởng

vừa là một nhà quản lý vừa là một lập trình viên, người đã thiết kế kiến trúc và bất kỳ phần quan trọng hoặc phức tạp nào của mã code. Các thành viên khác trong nhóm làm việc trên thiết kế chi tiết và mã hóa, dưới sự chỉ đạo của lập trình viên trưởng.

### chief programmer team 110 ( đội ngũ lập trình viên trưởng )

Nó bao gồm lập trình viên trưởng, người được hỗ trợ bởi lập trình viên dự phòng, thư ký lập trình và từ một đến ba lập trình viên. Khi cần thiết, nhóm được hỗ trợ bởi các chuyên gia trong các lĩnh vực khác, chẳng hạn như các vấn đề pháp lý hoặc tài chính... ( ví dụ như 1 ca phẫu thuật được chỉ đạo bởi 1 bác sĩ phẫu thuật trưởng thì cần phải có những bác sĩ phụ, y tá khác hỗ trợ )

### Ưu, nhược điểm của nhóm code có sếp kiểu cũ-kiểu cổ điển là gì ? (chief programmer team).?

- Ưu điểm :

+ Cần ít nhân lực (khoảng 6 người), mỗi người có 1 công việc được phân công rõ ràng  
=> Giảm chi phí, thời gian, đảm bảo chất lượng sản phẩm.

+ Quản lý nhân sự tốt hơn so với nhóm code dân chủ: vì có hệ thống cấp bậc và có người chịu trách nhiệm với dự án, theo dõi tiến độ

+ Lập trình viên chỉ cần lập trình, chứ không cần lo đến làm tài liệu vì đã có thư ký lo  
+ Khi sếp vắng mặt thì có sếp dự bị -> dự án vẫn được tiếp tục

+ Số kênh giao tiếp giảm so với nhóm dân chủ

**- Nhược điểm :**

+ Không thực tế do Chief Programmer là người phải vừa giỏi việc lập trình, vừa giỏi công việc quản lý. Nhưng thường người quản lý giỏi thì code kém và ngược lại do vậy rất khó để kiếm được người có khả năng như vậy.

+ Backup programmer là người cũng phải có khả năng như Chief tuy nhiên lại phải ở vị trí dự bị cho sếp và trả lương thấp hơn, do đó cũng rất khó kiếm ra người như vậy

+ Programming Secretary (thư ký lập trình) cũng khó tìm vì những lập trình viên thường rất dị ứng với công việc giấy tờ, do đó để tìm người lập trình chỉ đơn thuần làm công việc giấy tờ là rất khó.

+ Bị lạm phát lãnh đạo: thiếu người làm trực tiếp nhưng nhiều người làm lãnh đạo

**D**

**democratic team 109 (nhóm dân chủ)**

là một nhóm hoạt động vì một mục tiêu chung, không có lãnh đạo duy nhất, không có lập trình viên nào đang cố gắng thăng cấp lên cấp độ tiếp theo. Điều quan trọng là bản sắc của nhóm và sự tôn trọng lẫn nhau.

**Ưu, nhược điểm của nhóm code democratic là gì (nhóm code dân chủ )**

**- Ưu điểm :**

+ Do mô hình này 1 người luôn khuyến khích người khác tìm ra lỗi của mình nên việc tìm ra lỗi trong phần mềm sẽ nhanh hơn, chất lượng phần mềm được cải thiện.

+ Các thành viên nắm chắc phần code của mình

+ Khả năng code mạnh, nhất là giải quyết các dự án khó

+ Các cá nhân đều bình đẳng về chức vụ và đều có quyền đưa ra nhận xét, ý kiến; bỏ qua cái tôi cá nhân để vì mục đích chung

+ Phù hợp với môi trường nghiên cứu

**- Nhược điểm :**

+ Việc tự test code của mình thường không hiệu quả

+ Quản lý thường khó chấp nhận kiểu lập trình Ego Programming này

+ Những người có kinh nghiệm thường không thoải mái khi để những người non kinh nghiệm kiểm tra code của mình

+ Do không có sự cạnh tranh, thăng tiến nên thường thích hợp ở những môi trường nghiên cứu

+ Số giao tiếp nhiều

+ Khó khăn trong việc bỏ qua cái tôi

Vì vậy đội phải được phát triển một cách tự nhiên.

## E

### egoless programming (lập trình bình đẳng) 109 \*

- **Định nghĩa:** Là một phong cách lập trình áp dụng với các nhóm, trong đó các yếu tố cá nhân được giảm thiểu để chất lượng có thể được cải thiện.

- **Giải thích cụ thể:** Theo phong cách *egoless programming*:

- + các cá nhân đều bình đẳng về chức vụ và đều có quyền đưa ra nhận xét, ý kiến;
- + khi cá nhân gặp lỗi thì mọi người đặt cảm xúc cá nhân sang một bên để nhìn nhận, đánh giá một cách khách quan và mang tính xây dựng;
- + các module được cá nhân tự do phát triển, code, test;
- + cả nhóm cùng 1 mong muốn là nâng cao chất lượng của sản phẩm và sản phẩm được phát triển sẽ là thành quả của cả nhóm.

- **Lợi ích:**

- + Giải quyết được các dự án khó, khả năng code cao
- + Các thành viên đều nắm chắc được phần của mình

- **Khó khăn:**

- + Việc tự test code sẽ khó phát hiện lỗi (*Do các cá nhân tin tưởng vào phần mã của mình nên không thực sự cố gắng tìm ra lỗi, và cũng khó có cách nhìn khách quan như được người khác test*)
- + Quản lý nhân sự trong nhóm khó khăn (*vì tất cả đều bình đẳng, không có ai nắm vai trò quản lý chung; vì vậy, nhóm hoạt động theo cách này thường được tự phát triển, có thể là những người cùng làm việc hợp nhau rồi tự hình thành.*)
- + Số giao tiếp nhiều
- + Khó khăn trong việc bỏ qua cái tôi

## H

### hierarchy (thứ bậc) 111

Dựa trên ý tưởng nhóm bác sĩ của 1 ca phẫu thuật.

Bác sĩ phẫu thuật được hỗ trợ bởi các bác sĩ phẫu thuật khác, bác sĩ gây mê và nhiều y tá; chỉ đạo hành động của tất cả các thành viên khác của nhóm và chịu trách nhiệm về mọi khía cạnh của hoạt động.

Với ví dụ này, thứ bậc trong nhóm phát triển cũng mang ý nghĩa tương tự, sếp của cả nhóm điều phối, quản lý cũng như chịu trách nhiệm cho dự án. Bên dưới có sếp dự bị, thư ký lập trình và các lập trình viên.

## K

### KPA - key process area (lĩnh vực tiến trình) 119

Mỗi cấp độ trưởng thành đều có các KPAs riêng của nó, mỗi lĩnh vực trong số đó cần được giải quyết thỏa đáng trước khi một tổ chức có thể được coi là đã đạt được cấp độ trưởng thành đó.

- Ví dụ:

- Đối với cấp độ 2, cấp độ được quản lý, các KPA là nhân sự, giao tiếp và điều phối, môi trường làm việc, quản lý hiệu suất, đào tạo và phát triển và lương thưởng
- Ngược lại, KPA cho cấp độ 5, cấp độ tối ưu hóa, là cải tiến năng lực liên tục, điều chỉnh hoạt động của tổ chức và liên tục đổi mới lực lượng lao động.

L

### **librarian 112 (thủ thư)**

là chức danh mà một số tổ chức đặt cho các thư ký lập trình. Thư ký lập trình không phải là trợ lý văn thư bán thời gian mà là thành viên trung tâm có tay nghề cao, được trả lương cao, của nhóm lập trình viên trưởng. Thư ký lập trình chịu trách nhiệm duy trì dự án thư viện sản xuất, tài liệu của dự án. Điều này bao gồm danh sách mã nguồn, JCL và dữ liệu thử nghiệm. Các lập trình viên đã giao mã nguồn của họ cho thư ký, người chịu trách nhiệm chuyển đổi mã nguồn sang dạng máy đọc được, biên dịch, liên kết, tải, thực thi và chạy các trường hợp thử nghiệm.

P

### **pair programming 118 (Lập trình cặp)**

Hai lập trình viên làm việc cùng nhau trên một máy tính thực hiện nhiệm vụ và đảm bảo rằng tất cả các trường hợp thử nghiệm hoạt động chính xác. Họ luân phiên đánh máy sau mỗi 15 hoặc 20 phút và thay nhau kiểm tra lỗi sau đó tích hợp vào phiên bản hiện tại của sản phẩm. Lập trình giúp giảm thời gian thực hiện nhiệm vụ đồng thời giúp họ học hỏi và nâng cao trình độ lẫn nhau. Tuy nhiên lập trình cặp không phải lúc nào cũng hoạt động tốt với những cá nhân nhút nhát hoặc hổng hách, hoặc với hai lập trình viên thiếu kinh nghiệm.

### **programmer 112 (lập trình viên)**

là những người chỉ tập trung vào công việc lập trình.

### **programming secretary 112**

**Thư ký lập trình** là thành viên trung tâm có tay nghề cao, được trả lương cao, của nhóm lập trình viên. Thư ký lập trình chịu trách nhiệm duy trì thư viện sản xuất dự án, tài liệu của dự án. Điều này bao gồm danh sách mã nguồn, JCL và dữ liệu thử nghiệm. Các lập trình viên đã giao mã nguồn của họ cho thư ký, người chịu trách nhiệm chuyển đổi mã nguồn sang dạng máy đọc được, biên dịch, liên kết, tải, thực thi và chạy các trường hợp thử nghiệm. Các lập trình viên do đó không làm gì khác ngoài chương trình. Tất cả các khía cạnh khác trong công việc của họ đều do thư ký chương trình xử lý. (Vì thư ký lập trình duy trì thư viện sản xuất dự án, một số tổ chức đã sử dụng thủ thư chức danh.)

### **Tại sao vị trí secretary programmer cũng khó tuyển ?**

+ Vì thư ký lập trình cũng là những lập trình viên có kỹ năng cao, trình độ chuyên môn tốt, tuy nhiên công việc của họ trọng dự án hầu hết đều liên quan đến tài liệu của dự án

+ Mà thông thường hững thì lập trình viên thường không muốn làm liên quan đến tài liệu, giấy tờ quá nhiều rất dị ứng với công việc giấy tờ, hay có thể nói là thường không muốn làm giấy tờ mà họ mong muốn làm lập trình hơn.

+ Do đó, mặc dù được trả lương cao nhưng để tìm người lập trình chỉ đơn thuần làm công việc giấy tờ như thư ký là rất khó.

## S

### **specialization 111**

**Chuyên môn hóa:** Mỗi thành viên trong nhóm chỉ thực hiện những nhiệm vụ mà họ đã được đào tạo.

### **superprogrammer 113**

Siêu lập trình viên là lập trình viên giỏi, có thể viết mã cực kì tốt và nhanh chóng, từ đó cho kết quả đầu ra gấp 4 hoặc 5 lần so với một lập trình viên trình độ trung bình giỏi.

## T

### **task 118 (Nhiệm vụ)**

Một task là một đơn vị thực thi hoặc một đơn vị công việc. Khái niệm task thường là tiến trình, tiến trình nhẹ, luồng, bước, yêu cầu hoặc yêu cầu truy vấn

Cặp lập trình viên đầu tiên đưa ra các trường hợp thử nghiệm và sau đó đưa ra đoạn mã (nhiệm vụ) đó. Các quy trình Agile giải quyết vấn đề này bằng cách để một cặp lập trình viên trong nhóm vẽ các trường hợp thử nghiệm cho một nhiệm vụ và cặp lập trình viên kia cùng triển khai mã bằng các trường hợp thử nghiệm đó.

### **team 107 (Nhóm đội)**

Hầu hết các sản phẩm quá lớn để được hoàn thành bởi một chuyên gia phần mềm duy nhất trong thời hạn nhất định. Do đó, sản phẩm phải được giao cho một nhóm các chuyên gia được tổ chức như một đội. Ví dụ, hãy xem xét quy trình phân tích. Để xác định sản phẩm mục tiêu trong vòng 2 tháng, có thể cần giao nhiệm vụ cho ba chuyên gia phân tích được tổ chức thành một nhóm dưới sự chỉ đạo của trưởng phòng phân tích. Tương tự, nhiệm vụ thiết kế có thể được chia sẻ giữa các thành viên trong nhóm thiết kế.

### **team leader 114**

Là người quản lý chịu trách nhiệm về mặt kỹ thuật trong nhóm code có sếp kiểu mới : review toàn bộ code và hỗ trợ kỹ thuật cho các thành viên trong nhóm.

### **Team leader khác gì với backup programmer ?**

- Team leader là thành viên trong nhóm code có sếp kiểu mới còn backup programmer là thành viên của nhóm code có sếp kiểu cũ.
- Team leader chỉ giỏi về mặt kĩ thuật chứ không giỏi về mặt quản lí còn backup programmer giỏi về cả 2 mặt đó.



- Team leader tham gia trực tiếp vào công việc của cả nhóm còn backup programmer chỉ là người dự bị cho sếp chính.

#### **Team leader khác gì với secretary programming ?**

- Team leader là thành viên trong nhóm code có sếp kiểu mới còn secretary programming là thành viên của nhóm code có sếp kiểu cũ.
- Team leader chịu trách nhiệm review toàn bộ code và hỗ trợ kỹ thuật cho cả nhóm còn secretary programming chịu trách nhiệm về tài liệu của dự án.
- Team leader thường dễ tìm kiếm hơn **secretary programming** vì lập trình viên ghét việc phải làm tài liệu cả ngày.
- Team leader sẽ làm việc trực tiếp với các programmer với **secretary programming** sẽ làm việc trực tiếp với sếp của dự án.
- Team leader còn trực tiếp thiết kế, xây dựng module thay vì để các thành viên làm hết

#### **team manager (người quản lý đội nhóm) 114:**

người quản lý chịu trách nhiệm về tất cả các quyết định quản lý phi kỹ thuật và có được kiến thức về các kỹ năng kỹ thuật của từng lập trình viên trong nhóm trong các cuộc họp nhóm được lên lịch thường xuyên.

N

#### **Nhóm code có sếp kiểu mới (nhóm code kết hợp) thì áp dụng luật Miller như thế nào ?**

- requirement workflows: do sếp đảm nhiệm
- analysis workflows: do sếp đảm nhiệm
- design workflow: do team leader đảm nhiệm
- implementation workflow : do team leader và lập trình viên đảm nhiệm
- test workflow : do team leader phụ trách và các lập trình viên hỗ trợ nhau
- Trong nhóm code có sếp kiểu mới, để giảm bớt trách nhiệm của sếp, thì trong đó sếp chỉ quản lý các vấn đề phi kỹ thuật như là thu nhập, bình đẳng, năng lực của các thành viên, sếp sẽ tạo ra team leader để quản lý kỹ thuật: review toàn bộ code và hỗ trợ kỹ thuật cho các thành viên và sếp không review code nhưng khi họp có thể tham gia để hỗ trợ kỹ thuật cho các thành viên
- Và 1 sếp không quản lý quá 7 người trở lên mà chỉ quản lý 3-5 người để tăng độ hiệu quả

#### **Nhóm code có sếp kiểu mới (nhóm code kết hợp) thì phát huy được ưu điểm gì của nhóm code dân chủ ?**

Phát huy được ưu điểm :

- Các thành viên trong team đều nắm rất chắc phần code của mình
- Tính bình đẳng cao hơn so với nhóm kiểu cũ, các thành viên cùng cấp độ bình đẳng với nhau. Các module sẽ do team leader thiết kế.
- Khả năng code cao nhất là giải quyết dự án khó
- Tinh thần phát hiện và sửa lỗi cao