**CHƯƠNG 3: CƠ SỞ HẠ TẦNG KỸ THUẬT**

**GIỚI THIỆU:**

- Khi dự án lớn lên, kéo theo sự chậm trễ, theo quy luật Brooks – thêm người vào 1 dự án đang tiến hành chậm trễ sẽ càng trễ hơn. Vì thế ta sẽ sẽ dùng các phương pháp để đối phó với vấn đề này qua các phương pháp quản lý thông tin.

- Các phương pháp này cần phần mềm hỗ trọ, phần mềm tham gia hầu hết vào mọi quy trình xử lý. Tuy nhiên, sự tham gia quản lý của con người cũng rất quan trọng, vì vậy không nên quá lạm dụng sự tự động hóa của phần mềm mà cần linh hoạt, thông minh trong việc sửa dụng các phần mềm.

**MỘT DỰ ÁN CẦN GÌ?**

- Trang web:

* Chủ yếu là kênh truyền thông một chiều từ dự án ra công chúng.
* Có thể đóng vai trò là giao diện quản trị cho các công cụ khác của dự án.

- Danh sách thư:

* Thường là diễn đàn truyền thông sôi nổi nhất trong dự án và được coi là "phương tiện ghi chép".

- Quản lý phiên bản:

* Cho phép nhà phát triển quản lý các thay đổi mã nguồn một cách thuận tiện, bao gồm việc hoàn nguyên và "chuyển đổi thay đổi".
* Cho phép mọi người theo dõi sự thay đổi trong mã nguồn.

- Theo dõi lỗi:

* Cho phép nhà phát triển theo dõi công việc, phối hợp với nhau và lập kế hoạch cho các bản phát hành.
* Cho phép mọi người truy vấn tình trạng của lỗi và ghi lại thông tin (ví dụ: hướng dẫn tái tạo) về các lỗi cụ thể.
* Có thể được sử dụng để theo dõi không chỉ lỗi mà còn công việc, bản phát hành, tính năng mới, v.v.

- Chat thời gian thực:

* Nơi cho các cuộc thảo luận nhanh, trao đổi câu hỏi/đáp.
* Không luôn được lưu trữ hoàn toàn.

**DANH SÁCH THƯ – MAILING LIST:**

- Là một hệ thống gửi và nhận thư điện tử tự động giữa một nhóm người qua địa chỉ email chung. Người dùng có thể đăng ký tham gia vào danh sách thư để nhận và gửi thư điện tử đến tất cả mọi người trong nhóm. Mỗi lời nhắn được gửi đến địa chỉ email của danh sách sẽ được chuyển đến tất cả các thành viên đăng ký trong danh sách đó.

*“Đừng cố gắng quản lý danh sách thư bằng tay - hãy sử dụng phần mềm quản lý danh sách”*

- Phần mềm phản lý việc đăng ký, hủy đăng ký đối với người nhận thư, quản lý thư rác, …

- Đăng ký qua email và web:

* Người dùng khi đăng ký nên nhận ngay một thông báo chào mừng tự động, cung cấp thông tin về danh sách thư, cách tương tác với phần mềm danh sách thư, và (quan trọng nhất) cách hủy đăng ký.

- Chế độ đọc tóm tắt hoặc từng thư một:

* Người đăng ký có thể chọn nhận một email mỗi ngày, chứa tất cả hoạt động của danh sách trong ngày đó. Đối với những người theo dõi danh sách một cách lỏng lẻo mà không tham gia, chế độ tóm tắt thường là lựa chọn ưa thích, vì nó cho phép họ quét tất cả các chủ đề cùng một lúc và tránh sự làm phiền của email đến vào những thời điểm ngẫu nhiên.

- Tính năng kiểm duyệt:

* Kiểm duyệt để kiểm tra bài viết, đảm bảo chúng không phải là thư rác và nói về chủ đề, trước khi chúng được gửi đến toàn bộ danh sách. Mặc dù cần có sự tham gia của con người, nhưng phần mềm có thể làm cho quá trình này trở nên dễ dàng hơn.

- Giao diện quản trị:

* Cho phép quản trị viên dễ dàng loại bỏ các địa chỉ lỗi thời. Điều này trở nên cấp bách khi địa chỉ của một người nhận bắt đầu gửi tự động "Tôi không còn ở địa chỉ này nữa" trả lời lại danh sách mỗi khi có bài viết mới. (Một số phần mềm danh sách thư thậm chí có thể phát hiện điều này tự động và tự động hủy đăng ký người đó.)

- Chỉnh sửa đầu trang:

* Nhiều người dùng có các quy tắc lọc và trả lời phức tạp được thiết lập trong chương đọc thư của họ. Phần mềm danh sách thư có thể thêm và chỉnh sửa một số đầu trang tiêu chuẩn để họ có thể tận dụng được (chi tiết hơn ở dưới).

- Lưu trữ trực tuyến:

* Tất cả các bài viết đến danh sách đều được lưu trữ và có sẵn trên web; hoặc một số phần mềm danh sách thư cung cấp giao diện đặc biệt để kết nối với công cụ lưu trữ ngoại vi như MHonArc (<http://www.mhonarc.org/>).

**CHỐNG THƯ RÁC**

*4.1. Lọc bài viết*

Có ba kỹ thuật cơ bản để ngăn chặn bài viết thư rác, và hầu hết các phần mềm danh sách thư đều cung cấp cả ba. Chúng tốt nhất khi được sử dụng cùng nhau:

- Chỉ tự động cho phép bài viết từ người đăng ký danh sách.

+ Để quản lý một danh sách gửi hiệu quả, hãy xem xét việc chỉ cho phép bài viết từ người đăng ký, nhưng đừng tự động loại bỏ bài viết từ người không đăng ký. Thay vào đó, hãy chuyển chúng đến giai đoạn kiểm duyệt để hỗ trợ câu hỏi hoặc đề xuất chân thành từ những người không đăng ký. Đồng thời, hãy cẩn thận với người đăng ký có thể sử dụng địa chỉ email khác nhau.

- Lọc bài viết qua phần mềm lọc thư rác.

+ Sử dụng phần mềm chống thư rác để giảm công việc kiểm duyệt. Mặc dù không có bộ lọc nào hoàn hảo, nhưng nó giúp giảm thiểu thư rác đáng kể. Tham khảo tài liệu của phần mềm danh sách gửi để biết cách cài đặt, và xem xét việc sử dụng công cụ như SpamAssassin hoặc SpamProbe.

- Có người kiểm duyệt.

+ Thực hiện giai đoạn kiểm duyệt cho những bài viết không được phép tự động do không phải từ người đăng ký hoặc đã được lọc qua phần mềm chống thư rác. Điều này bao gồm xem xét của con người, với tùy chọn xác nhận một bài viết cụ thể hoặc cho phép tất cả các bài viết tương lai từ người gửi đó để giảm công việc kiểm duyệt sau này.

*4.2. Giấu địa chỉ:*

Để ngăn danh sách thư của bạn trở thành nguồn địa chỉ cho máy gửi thư rác, có một kỹ thuật phổ biến là làm cho các địa chỉ email của người dùng trở nên khó nhìn thấy đối với các máy gửi này, ví dụ như:

thay thế [jrandom@somedomain.com](mailto:jrandom@somedomain.com)

thành jrandom\_AT\_somedomain.com

hoặc [jrandomNOSPAM@somedomain.com](mailto:jrandomNOSPAM@somedomain.com)

Mục tiêu là làm cho địa chỉ email trở nên khó đọc đối với máy thu thập thư rác, ngăn chúng từ việc lấy địa chỉ trên các trang web, bao gồm cả các bản lưu trữ của danh sách thư.

Cách dùng này có thể gây ra tác dụng không mong muốn, người dùng sẽ khó khăn hơn khi phải nhớ địa chỉ email của họ khi thêm các kí tự đặc biệt vào, và phương pháp này cũng trở nên lỗi thời và kém hiệu quả khi các công nghệ thu thập email để gửi thư rác ngày càng thông minh hơn.

**IDENTIFICATION AND HEADER MANAGEMENT**

các phương pháp hay nhất để quản lý danh sách gửi thư và cung cấp hướng dẫn cho người đăng ký về cách sắp xếp và lọc email hiệu quả. Nó nhấn mạnh việc sử dụng các tiêu đề email, cụ thể là tiêu đề "Tới", để dễ dàng lọc và sắp xếp các email trong danh sách gửi thư.

Có thể tiềm ẩn của việc tự động thêm tiền tố vào tiêu đề Chủ đề, cảnh báo về nguy cơ đẩy thông tin quan trọng ra khỏi màn hình..

các tiêu đề tiêu chuẩn cụ thể, chẳng hạn như "list-help", "list-unsubscribe" và "list-post" thường được sử dụng trong danh sách gửi thư. Những tiêu đề này được cung cấp bởi phần mềm quản lý danh sách gửi thư và có thể được sử dụng cho mục đích quản trị.

Tham chiếu đến các URL như "http://www.nisto.com/listspec/list-manager-intro.html" và "http://www.faqs.org/rfcs/rfc2369.html" cung cấp bổ sung và chi tiết thông số kỹ thuật có thể được tìm thấy tại các liên kết đó.

**THE GREAT REPLY-TO DEBATE**

giải quyết cuộc tranh luận xung quanh việc liệu phần mềm danh sách gửi thư có nên tự động đặt tiêu đề "Trả lời" để chuyển hướng các câu trả lời đến danh sách gửi thư hay không. Tiêu đề "Trả lời" thường cho biết địa chỉ email mà thư trả lời sẽ được chuyển đến.

Việc tự động đặt tiêu đề "Trả lời" cho địa chỉ danh sách là nó khuyến khích sự cộng tác và đảm bảo rằng các câu trả lời được chuyển hướng đến toàn bộ danh sách theo mặc định. Điều này phù hợp với mục tiêu duy trì các cuộc thảo luận trong diễn đàn công cộng của danh sách gửi thư.

Tuy nhiên, văn bản nêu ra hai nhược điểm đáng kể liên quan đến phương pháp này. Đầu tiên là vấn đề "Không thể tìm đường về nhà", trong đó một số người dùng có thể cần đặt địa chỉ email "thực" của họ trong trường "Trả lời" do cấu hình email duy nhất hoặc các hạn chế tại nơi làm việc. Nếu phần mềm danh sách ghi đè điều này, người gửi ban đầu có thể bỏ lỡ phản hồi.

Nhược điểm thứ hai liên quan đến kỳ vọng và hành vi của người dùng. Người dùng email có kinh nghiệm đã quen với hai phương thức trả lời cơ bản: "trả lời tất cả" và "trả lời tác giả". Việc ghi đè "Trả lời" của người gửi ban đầu có thể dẫn đến hành vi không mong muốn, trong đó người dùng vô tình gửi tin nhắn riêng tư vào danh sách công khai.

**TWO FANTASIES**

1. Triển khai Khóa trả lời danh sách:

- trong tương lai trong đó người đọc thư triển khai khóa "trả lời danh sách" chuyên dụng. Khóa này sẽ tự động xác định địa chỉ danh sách gửi thư bằng cách sử dụng tiêu đề danh sách tùy chỉnh được đề cập trước đó trong văn bản. Khi người dùng nhấn phím này, thư trả lời sẽ được gửi trực tiếp đến danh sách gửi thư, ngoại trừ mọi địa chỉ người nhận khác. Đây được coi là một giải pháp tiềm năng để đơn giản hóa cuộc tranh luận về cài đặt "Trả lời".

2. Ưu tiên của mỗi người đăng ký khi trả lời tin nhắn:

- Đó là việc trộn lẫn "Trả lời" trở thành tùy chọn của mỗi người đăng ký. Trong trường hợp này, người đăng ký có thể chọn riêng lẻ xem họ muốn đặt "Trả lời" trong danh sách (trên bài đăng của cả người khác hoặc bài đăng của chính họ) hay họ muốn để "Trả lời" một mình.

- chưa có phần mềm quản lý danh sách nào cung cấp khả năng tùy chỉnh cho mỗi người đăng ký này, Hiện nay danh sách gửi thư thường được đặt cấu hình với cài đặt chung cho hành vi "Trả lời".

**ARCHIVING**

Phần mềm quản lý danh sách gửi thư là một loại phần mềm được sử dụng để tổ chức, quản lý, và tự động hóa quá trình gửi và nhận thư qua danh sách gửi thư. Những phần mềm này giúp quản trị viên và người sử dụng dễ dàng quản lý các danh sách gửi thư, thực hiện các hoạt động quản lý, và tối ưu hóa trải nghiệm tham gia của người dùng.

Các chức năng chính của phần mềm quản lý danh sách gửi thư bao gồm:

1. Quản lý Người Dùng:

- Đăng ký và hủy đăng ký thành viên từ danh sách gửi thư.

- Quản lý danh sách thành viên, bao gồm thêm, sửa, xóa thông tin thành viên.

2. Gửi và Nhận Thư:

- Gửi thư đến tất cả hoặc một phần của danh sách gửi thư.

- Quản lý quyền lợi và hạn chế của người sử dụng, ví dụ như việc chỉ cho phép người đăng ký mới nhận thư sau khi xác nhận đăng ký.

3. Quản lý Thông Tin:

- Lưu trữ và quản lý lịch sử các thư gửi và nhận.

- Tạo và quản lý các thông báo và thông điệp tự động.

4. Bảo mật và Quyền Lợi:

- Bảo vệ thông tin cá nhân của thành viên.

- Quản lý quyền lợi và quyền truy cập của người sử dụng đối với các chức năng của danh sách gửi thư.

5. Thống Kê và Theo Dõi:

- Cung cấp các báo cáo và thống kê về hoạt động của danh sách gửi thư.

- Theo dõi hiệu suất của các chiến dịch gửi thư.

6. Quản lý Phiên Bản và Lịch Sử:

- Hỗ trợ quản lý các phiên bản của danh sách gửi thư.

- Lưu trữ lịch sử thay đổi và cập nhật của danh sách.

**SOFTWARE**

danh sách các công cụ nguồn mở để quản lý và lưu trữ danh sách, có thể hữu ích cho các dự án yêu cầu thiết lập danh sách gửi thư.

Phần mềm quản lý danh sách gửi thư:

1. **Mailman**: [http://www.list.org/]

(http://www.list.org/) (Bao gồm trình lưu trữ tích hợp sẵn và hỗ trợ các trình lưu trữ bên ngoài.)

1. **SmartList**: [http://www.procmail.org/]

(http://www.procmail.org/) (Được thiết kế để sử dụng với hệ thống xử lý thư Procmail.)

3. **Ecartis**: [http://www.ecartis.org/](http://www.ecartis.org/)

4. **ListProc**:[http://listproc.sourceforge.net/](http://listproc.sourceforge.net/)

5. **Ezmlm**: [http://cr.yp.to/ezmlm.html]

(http://cr.yp.to/ezmlm.html) (Được thiết kế để hoạt động với hệ thống gửi thư Qmail.)

6. **Dada**: [http://mojo.skazat.com/]

(http://mojo.skazat.com/) (Mặc dù trang web đã cố gắng che giấu nhưng đây là phần mềm miễn phí được phát hành theo GNU General Giấy phép Công cộng và có trình lưu trữ tích hợp.)

Phần mềm lưu trữ danh sách gửi thư:

1. **Mhonarc**:[http://www.mhonarc.org/](http://www.mhonarc.org/)

2. **Hypermail**:[http://www.hypermail.org/](http://www.hypermail.org/)

3. **Lurker**: [http://sourceforge.net/projects/lurker/](http://sourceforge.net/projects/lurker/)

4. **Procmail**:[http://www.procmail.org/](http://www.procmail.org/) (Phần mềm đồng hành với SmartList; một hệ thống xử lý thư chung có thể được định cấu hình làm bộ lưu trữ .)

**VERSION CONTROL (QUẢN LÝ PHIÊN BẢN)**

Hệ thống quản lý phiên bản (hoặc hệ thống quản lý biên dịch) là sự kết hợp giữa công nghệ và thực tiễn để theo dõi và kiểm soát các thay đổi đối với tệp của dự án, đặc biệt là mã nguồn, tài liệu và trang web.

**Version Control Vocabulary**

*Commit*: trong phát triển phần mềm có nghĩa là lưu trữ một thay đổi vào cơ sở dữ liệu quản lý phiên bản sao cho nó có thể được tích hợp vào các phiên bản phần mềm trong tương lai. “Commit” có thể được sử dụng như một động từ hoặc một danh từ. Như một danh từ, nó tương đương với “thay đổi”.

*Log messages*: là một trong những tài liệu quan trọng nhất trong bất kỳ dự án nào: chúng là cầu nối giữa ngôn ngữ kỹ thuật cao của các thay đổi mã lệnh riêng lẻ và ngôn ngữ hướng người dùng hơn của các tính năng, sửa lỗi và tiến độ dự án.

*Update*: Yêu cầu những thay đổi của người khác được đưa vào bản sao dự án của bạn. Đây là một hoạt động rất phổ biến; hầu hết các nhà phát triển đều cập nhật mã của họ vài lần trong ngày để biết rằng họ đang chạy một phiên bản gần giống với các nhà phát triển khác và để nếu họ thấy lỗi, họ có thể đảm bảo rằng lỗi này chưa được sửa.

*Repository (kho):* Một số hệ thống quản lý phiên bản được tập trung: có một kho lưu trữ chính, lưu trữ tất cả các thay đổi đối với dự án. Một số khác được phân cấp: mỗi nhà phát triển có kho lưu trữ riêng và các thay đổi có thể được hoán đổi qua lại giữa các kho lưu trữ một cách tùy ý. Hệ thống kiểm soát phiên bản theo dõi sự phụ thuộc giữa các thay đổi và khi đến thời điểm phát hành, một tập hợp cụ thể các thay đổi sẽ được phê duyệt cho bản phát hành đó.

*Checkout*: Quá trình lấy bản sao của dự án từ kho lưu trữ. Quá trình kiểm tra thường tạo ra một cây thư mục được gọi là "bản sao làm việc", từ đó các thay đổi có thể được đưa trở lại kho lưu trữ ban đầu. Trong một số hệ thống kiểm soát phiên bản phi tập trung, mỗi bản sao làm việc tự nó là một kho lưu trữ và các thay đổi có thể được đẩy ra (hoặc kéo vào) bất kỳ kho lưu trữ nào sẵn sàng chấp nhận chúng.

*working copy*: Cây thư mục riêng của nhà phát triển chứa các tệp mã nguồn của dự án và có thể là các trang web hoặc tài liệu khác. Một bản sao làm việc cũng chứa một ít siêu dữ liệu được quản lý bởi hệ thống quản lý phiên bản, cho bản sao làm việc biết nó đến từ kho lưu trữ nào.

*revision, change, changeset:* "Bản sửa đổi" thường là một phiên bản cụ thể của một tệp hoặc thư mục cụ thể. Một số hệ thống cũng sử dụng "bản sửa đổi", "thay đổi" hoặc "bộ thay đổi" để chỉ đến một tập hợp các thay đổi được cam kết cùng nhau như một đơn vị khái niệm.

*diff:* là biểu diễn văn bản của một sự thay đổi. Nó cho thấy những dòng nào đã được thay đổi và thay đổi như thế nào. Một nhà phát triển đã quen thuộc với mã nguồn có thể đọc được một diff và hiểu thay đổi đó đã làm gì và thậm chí có thể tìm ra lỗi.

*tag:* nhãn được gán cho một bộ sưu tập các tệp cụ thể tại các phiên bản xác định. Thông thường nhãn được để lưu giữ các bản chụp thú vị của dự án.

*Branch:* Một bản sao của dự án, dưới sự kiểm soát phiên bản nhưng bị cô lập, để những thay đổi được thực hiện đối với nhánh không ảnh hưởng đến phần còn lại của dự án và ngược lại, trừ khi các thay đổi được "hợp nhất" một cách có chủ ý từ bên này sang bên kia

*Merge:* Để di chuyển một sự thay đổi từ nhánh này sang nhánh khác. Điều này bao gồm việc hợp nhất từ ​​nhánh chính sang một số nhánh khác hoặc ngược lại.

*Conflict:* Điều này xảy ra khi hai người cố gắng thực hiện những thay đổi khác nhau ở cùng một vị trí trong mã. Tất cả các hệ thống kiểm soát phiên bản đều tự động phát hiện xung đột và thông báo cho ít nhất một trong số những người có liên quan rằng những thay đổi của họ xung đột với những thay đổi của người khác.

*Lock:* Một cách để tuyên bố ý định độc quyền thay đổi một tập tin hoặc thư mục cụ thể.

**Choosing a Version Control System**

Theo văn bản này, hai hệ thống kiểm soát phiên bản phổ biến nhất trong thế giới phần mềm miễn phí là Concurrent Versions System và Subversion CVS đã có 1 thời gian dài, nhiều nhà phát triển có kinh nghiệm đã quen với nó. Tuy nhiên nó có 1 số nhược điểm. Nó không cung cấp cách dễ dàng để tham khảo các thay đổi của nhiều tệp, không cho phép đổi tên hoặc sao chép các file được quản lý, hỗ trọ hợp nhất kém, tốc độ hoạt động chậm khi có nhiều file.Trong những năm qua, SVN đã lấy lại được chỗ đứng. Đã có nhiều phiên bản hệ thống quản lý phiên bản mới miễn phí xuất hiện.

**Using the Version Control System**

**Version everything**

Đảm bảo không chỉ quản lý mã nguồn của dự án, mà còn quản lý các trang web, tài liệu, câu hỏi thường gặp, ghi chú thiết kế và bất kỳ thông tin nà có thể chỉnh sửa.

Lý do quản lý tất cả thông tin trong một nơi kết hợp là để người dùng chỉ cần học một chức năng duy nhất cho các bài đăng.

Không quản lý các tập tin được tạo ra khi quay, do chúng không phải là dữ liệu được chỉnh sửa thực tế, do chúng được tạo ra từ các tập tin khác.

**BROWSABILITY**

Kho lưu trữ của dự án nên có khả năng duyệt được trên Web. Điều này không chỉ là khả năng xem các bản sửa i mới nhất của dự án mà còn có thể quay ngược thời gian và xem các bản sửa đổi trước đó, xem sự khác biệt giữa các bản sửa đổi, đọc thông báo tường trình cho các thay đổi đã chọn.

Khả năng duyệt rất quan trọng vì đây là một cổng nhẹ vào dữ liệu của dự án.

Khả năng duyệt cũng có nghĩa với các URL chuẩn để xem các bản sửa đổi cụ thể của tệp và để xem bản sửa đổi mới nhất tại bất kỳ thời điểm nào.

Một số hệ thống kiểm soát phiên bản đi kèm với cơ chế duyệt kho lưu trữ được tích hợp sẵn, trong khi các hệ thống khác dựa vào các công cụ của bên thứ ba để thực hiện việc đó. Ba công cụ như: ViewCVS, CVSWeb và WebSVN. Cái đầu tiên hoạt động với cả CVS và Subversion, cái thứ hai chỉ với CVS và cái thứ ba chỉ với Subversion.

**COMMIT EMAILS**

Mọi cam kết đối với kho lưu trữ sẽ tạo một email cho biết ai đã thực hiện thay đổi, họ thực hiện thay đổi khi nào, tệp và thư mục nào đã thay đổi cũng như cách chúng thay đổi.

Các chi tiết cụ thể về việc thiết lập email cam kết sẽ khác nhau tùy thuộc vào hệ thống kiểm soát phiên bản của bạn, nhưng thông thường sẽ có một tập lệnh hoặc tiện ích đóng gói khác để thực hiện việc đó.

**USE BRANCHES TO AVOID BOTTLENECKS(SỬ DỤNG NHIỀU NHÁNH ĐỂ TRÁNH TẮC NGHẼN)**

* Những người không chuyên về việc sử dụng hệ thống quản lý phiên bản thường có một chút lo lắng khi sử dụng branching (phân nhánh)và merging(nhập nhánh), có lẽ do giao diện của một số hệ thống như CVS khá phức tạp. Tuy nhiên, quan trọng là vượt qua nỗi sợ hãi này và học cách sử dụng chúng một cách hiệu quả. Branching(phân nhánh) và merging(nhập nhánh), không phức tạp như có vẻ và trở nên quan trọng hơn khi một dự án phát triển.
* Branches(nhánh) rất quan trọng vì chúng cho phép các nhà phát triển làm việc trong các khu vực riêng biệt, biến một tài nguyên hạn chế (không gian làm việc trong mã nguồn dự án) thành một tài nguyên dồi dào. Merging(nhập nhánh) quan trọng vì nó đảm bảo rằng một thay đổi chỉ được commit một lần và định danh duy nhất của nó được ghi lại trong hệ thống quản lý phiên bản. Điều này ngăn chặn sự trùng lặp của các thay đổi và giúp quản lý sách kế toán và quản lý phiên bản một cách chính xác.
* Branching cho phép các nhà phát triển làm việc trên các tính năng hoặc sửa lỗi mới mà không làm phiền công việc của người khác. Nó cũng cho phép cô lập mã nguồn để sửa lỗi hoặc ổn định một phiên bản mà không cần lo lắng về các thay đổi trong quá trình phát triển.
* Quan trọng là sử dụng branches một cách tự do, nhưng mỗi branch(nhánh) chỉ nên hoạt động trong khoảng thời gian cần thiết. Các branch hoạt động đòi hỏi sự chú ý từ cộng đồng phát triển, vì vậy quan trọng là merge các thay đổi trở lại main line và loại bỏ các branch không cần thiết.

**SINGULARITY OF INFORMATION(TÍNH ĐƠN NHẤT CỦA THÔNG TIN)**

* Quy tắc quan trọng khi sử dụng hệ thống quản lý phiên bản là không bao giờ commit một thay đổi hai lần. Một thay đổi cụ thể chỉ nên xuất hiện trong hệ thống quản lý phiên bản một lần duy nhất, và phiên bản (hoặc tập hợp các phiên bản) mà thay đổi đó xuất hiện trở thành định danh duy nhất của nó từ thời điểm đó trở đi.
* Khi muốn thêm một tính năng mới nhưng không thể thuyết phục mọi người khác về lợi ích của nó, việc tạo branch cho phép người đó thử nghiệm ở một góc tách biệt. Nếu thành công, có thể mời các nhà phát triển khác kiểm tra kết quả và sau đó merge tính năng đó từ nhánh gốc sang nhánh chính.
* Người dùng phiên bản kiểm soát không chuyên sâu thường sợ hãi việc tạo và merge branch, có thể là do giao diện của CVS khá khó hiểu. Tuy nhiên, khả năng tạo và merge branch là quan trọng để phát triển cộng tác và quản lý mã nguồn hiệu quả.
* Khi thực hiện merge, tránh việc lặp lại thông điệp log của thay đổi gốc. Thay vào đó, chỉ cần chỉ ra đây là một merge và cung cấp số phiên bản nhận dạng của thay đổi gốc, với tóm tắt tác động trong một câu. Điều này giúp tránh sự nhầm lẫn trong quản lý phiên bản và giữ cho thông điệp log được duy trì một cách chính xác."
* Thông điệp log trong mã nguồn mở (open source) là một bản ghi mô tả các thay đổi được thực hiện trên mã nguồn. Mỗi khi một nhà phát triển thực hiện một thay đổi, như thêm một tính năng mới, sửa lỗi, hoặc thay đổi cấu trúc mã nguồn, họ thường đi kèm với một thông điệp log.
* Thông điệp log này thường chứa các thông tin như:
* Mục đích của thay đổi: Mô tả tại sao thay đổi này được thực hiện, nó giải quyết vấn đề gì hoặc thêm vào dự án những gì.
* Cách thay đổi được thực hiện: Mô tả cách thức thay đổi được triển khai, đôi khi bao gồm các hướng dẫn chi tiết cho những người khác muốn hiểu rõ cách thay đổi được thực hiện.
* Thay đổi liên quan: Nếu có các thay đổi liên quan hoặc phụ thuộc, thông điệp log có thể đề cập đến chúng.
* Tác giả của thay đổi: Thông tin về người thực hiện thay đổi.
* Ngày thực hiện: Thời điểm thay đổi được commit vào hệ thống kiểm soát phiên bản.
* Thông điệp log giúp người khác hiểu rõ quá trình phát triển và giữ cho lịch sử của dự án được theo dõi một cách chi tiết. Nó là một phần quan trọng của quy trình làm việc của nhóm phát triển và hỗ trợ trong việc duy trì và quản lý mã nguồn mở.

**AUTHORIZATION(ỦY QUYỀN)**

* Hầu hết các hệ thống quản lý phiên bản cung cấp tính năng cho phép kiểm soát quyền commit của người dùng trong các phần cụ thể của kho lưu trữ. Nhiều dự án thường sử dụng tính năng này để kiểm soát chặt chẽ quyền truy cập, đặc biệt là chỉ cho phép commit ở những khu vực được phê duyệt. Tuy nhiên, việc sử dụng hệ thống kiểm soát chặt có thể tạo ra sự cứng nhắc không cần thiết.
* Một số dự án chọn sử dụng hệ thống trên tinh thần tôn trọng và tin tưởng, nơi khi một người được cấp quyền commit cho một phần cụ thể, họ thực tế nhận được mật khẩu cho phép họ commit ở bất kỳ nơi nào trong dự án. Mặc dù có rủi ro nhỏ, nhưng trong môi trường dự án tích cực, mọi commit đều được xem xét và người khác sẽ lưu ý nếu có ai đó commit sai quy định. Việc quay lại và hủy bỏ một thay đổi là khá đơn giản vì tất cả đều được quản lý trong hệ thống kiểm soát phiên bản.
* Việc tiếp cận thoải mái này mang lại nhiều ưu điểm, bao gồm sự linh hoạt khi mở rộng quyền truy cập và khuyến khích sự tôn trọng và tin tưởng trong dự án. Tuy nhiên, quy định cần được duy trì để đảm bảo người dùng chỉ commit ở những khu vực mà họ có chuyên môn. Tính minh bạch và sự tôn trọng đối với người commit mới giúp xây dựng một môi trường làm việc tích cực.

**BUG TRACKER(THEO DÕI LỖI)**

* Hệ thống quản lý bug thường được sử dụng để theo dõi yêu cầu tính năng mới, công việc một lần, bản vá không yêu cầu—thực sự là bất cứ điều gì có các trạng thái bắt đầu và kết thúc rõ ràng, với các trạng thái chuyển đổi tùy chọn ở giữa, và tích luỹ thông tin qua thời gian. Vì lý do này, các hệ thống quản lý bug còn được gọi là hệ thống theo dõi vấn đề, hệ thống theo dõi lỗi, hệ thống theo dõi tác phẩm, hệ thống theo dõi yêu cầu, hệ thống quản lý vấn đề v.v.
* Trong cuốn sách này, tôi sẽ tiếp tục sử dụng thuật ngữ 'bug tracker' để chỉ phần mềm thực hiện theo dõi, vì đó là cái mà hầu hết mọi người gọi nó, nhưng sẽ sử dụng thuật ngữ 'issue' để chỉ một mục duy nhất trong cơ sở dữ liệu của bug tracker. Điều này giúp chúng ta phân biệt giữa hành vi hoặc hành vi không mong muốn mà người dùng gặp phải (nghĩa là, lỗi), và hồ sơ của bug tracker về việc phát hiện lỗi, chẩn đoán và giải quyết cuối cùng của nó. Hãy nhớ rằng mặc dù hầu hết các vấn đề là về lỗi thực sự, nhưng chúng cũng có thể được sử dụng để theo dõi các loại công việc khác.
* Vòng đời cơ bản của một vấn đề trông như sau
* Người nào đó đề xuất vấn đề: Họ cung cấp một tóm tắt, mô tả ban đầu (bao gồm cả một bản mô phỏng, nếu có thể), và bất kỳ thông tin nào mà bug tracker yêu cầu.
* Người khác đọc vấn đề, thêm bình luận và có thể yêu cầu làm rõ từ người đề xuất.
* Lỗi được tái tạo: Một ai đó khác ngoài người đề xuất có thể làm cho lỗi xảy ra, chứng minh rằng đó là lỗi thực sự và xác nhận với người đề xuất rằng họ đã đóng góp vào dự án bằng cách báo cáo một lỗi thực sự.
* Chẩn đoán lỗi: Nguyên nhân được xác định, và nếu có thể, công sức cần thiết để sửa lỗi được ước tính. Điều này cần được ghi lại trong vấn đề để nếu người chẩn đoán bất ngờ phải rời khỏi dự án, người khác có thể tiếp tục.
* Vấn đề được lên lịch giải quyết: Quyết định khi nào lỗi sẽ được sửa. Đôi khi chỉ đơn giản là quyết định lỗi nên được sửa trong bản phát hành tương lai nào đó, hoặc quyết định rằng lỗi không cần phải chặn bất kỳ bản phát hành cụ thể nào.
* Lỗi được sửa: Sửa lỗi hoặc thực hiện công việc, áp dụng bản vá, v.v. Thay đổi hoặc bộ thay đổi giải quyết vấn đề được ghi lại trong một bình luận trên vấn đề. Sau đó, vấn đề có thể được đóng hoặc đánh dấu là đã giải quyết.
* Các biến thể phổ biến:
* Đóng ngay sau khi đề xuất: Nếu lỗi không phải là lỗi hoặc là sự hiểu lầm từ phía người dùng, vấn đề có thể được đóng ngay sau khi đề xuất.
* Đóng làm trùng lặp: Khi người nào đó báo cáo một vấn đề đã biết đến, nó có thể được đóng làm trùng lặp, đặc biệt là nếu đó là một lỗi đã được giải quyết trước đó.
* Phục hồi vấn đề: Người chịu trách nhiệm có thể phục hồi vấn đề sau khi đóng nếu họ không thể tái tạo lỗi hoặc nếu họ phản đối bản vá được áp dụng.
* quản lý bug không chỉ là về việc sửa lỗi mà còn liên quan đến việc quản lý các nhiệm vụ khác như yêu cầu tính năng, nhiệm vụ một lần và thậm chí bản vá không yêu cầu. Vòng đời của mỗi vấn đề cụ thể có thể thay đổi, nhưng hình dạng cơ bản là giống nhau, và nó có ảnh hưởng đối với cách dự án mã nguồn mở sử dụng hệ thống quản lý bug của mình."

**INTERACTION WITH MAILING LISTS (TƯƠNG TÁC VỚI DANH SÁCH GỬI THƯ)**

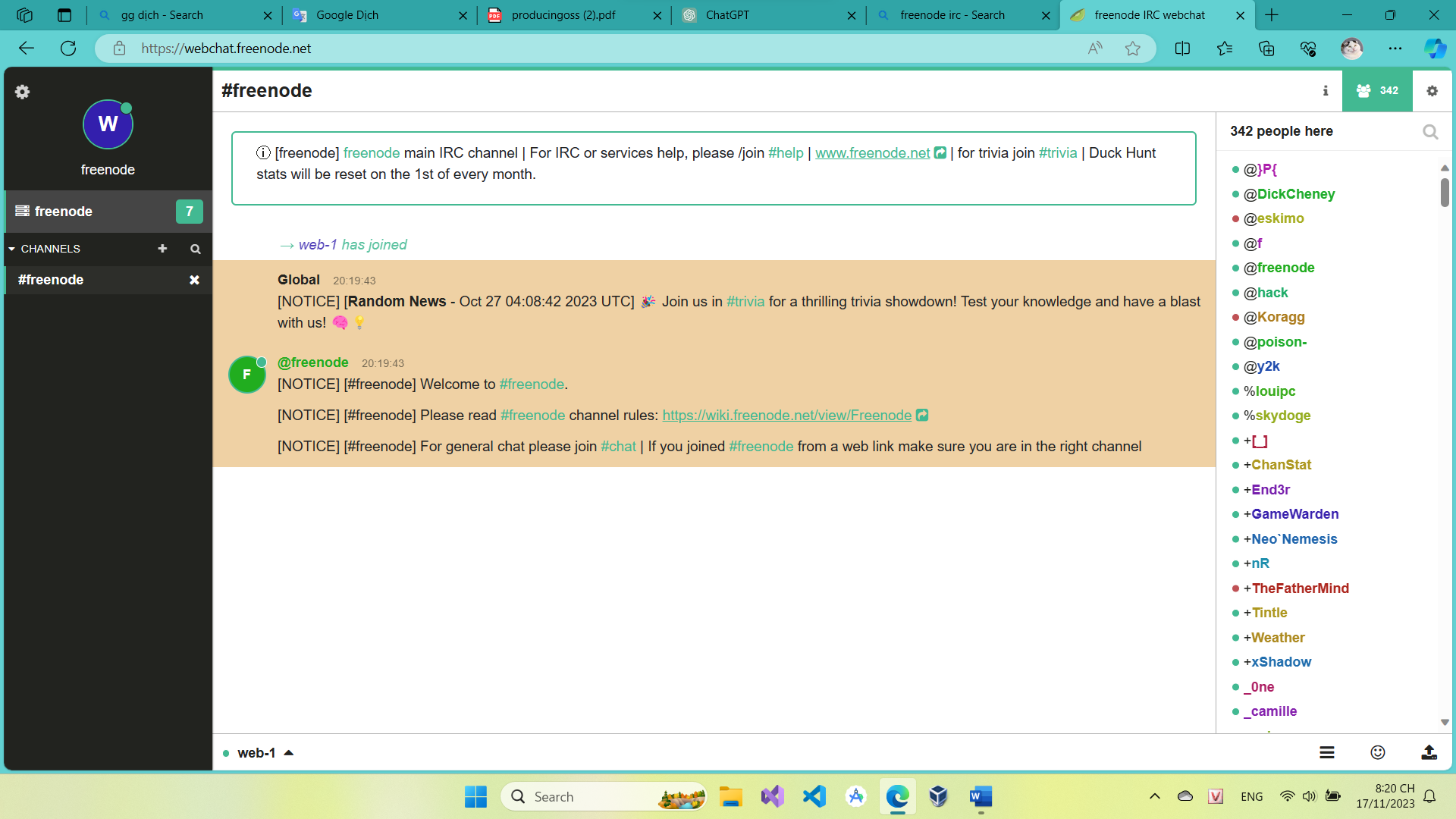
* bug tracker không trở thành diễn đàn thảo luận thời gian thực nó không phải là nơi thích hợp để thảo luận thời gian thực. Hãy nghĩ về nó như một bảo quản thông tin, một cách để tổ chức sự thật và tham khảo đến các cuộc thảo luận khác, chủ yếu là những cuộc thảo luận diễn ra trên danh sách gửi thư.
* Có hai lý do để tạo ra sự phân biệt này. Thứ nhất, bug tracker không thuận tiện bằng danh sách gửi thư (hoặc so với diễn đàn thảo luận thời gian thực, ví dụ như chat). Điều này không phải là do bug tracker có thiết kế giao diện người dùng kém, mà chỉ là giao diện của chúng được thiết kế để chụp và hiển thị các trạng thái rời rạc, không phải thảo luận tự do.
* Thứ hai, không phải ai cũng muốn tham gia vào cuộc thảo luận về một vấn đề cụ thể trên bug tracker. Quản lý vấn đề tốt (xem phần "Chia sẻ Nhiệm vụ Quản lý cũng như Nhiệm vụ Kỹ thuật" trong Chương 8, Quản lý Tình nguyện viên) đòi hỏi đảm bảo rằng mỗi vấn đề đều được đưa đến sự chú ý của những người đúng, thay vì yêu cầu mỗi nhà phát triển đều theo dõi tất cả các vấn đề. Trong phần "Không có Cuộc trò chuyện trong Bug Tracker" ở Chương 6, Giao tiếp, chúng ta sẽ xem cách để đảm bảo rằng mọi người không vô tình chuyển cuộc thảo luận ra khỏi các diễn đàn thích hợp và vào bug tracker.
* Một số bug tracker có thể theo dõi danh sách gửi thư và tự động đăng nhập tất cả các email liên quan đến một vấn đề đã biết. Thông thường, điều này được thực hiện bằng cách nhận diện số nhận dạng của vấn đề trong dòng chủ đề của thư, là một phần của một chuỗi đặc biệt; nhà phát triển học cách bao gồm những chuỗi này trong thư của họ để thu hút sự chú ý của bug tracker. Bug tracker có thể lưu toàn bộ email, hoặc (tốt hơn) chỉ ghi lại một liên kết đến thư trong lưu trữ thư thông thường. Dù cách nào đi nữa, đây là một tính năng rất hữu ích; Nếu bug tracker của bạn có tính năng này, hãy đảm bảo bật nó và nhắc nhở mọi người sử dụng nó.
* Nếu một email liên quan đến vấn đề đã biết được gửi đến, bug tracker có thể tự động đăng nhập vào email đó và liên kết nó với ticket tương ứng. Việc này giúp cho quá trình quản lý và giải quyết lỗi trở nên dễ dàng và hiệu quả hơn.

**PRE-FILTERING THE BUG TRACKER (TIỀN LỌC BUG TRACKER)**

* Những thách thức phổ biến trong quản lý cơ sở dữ liệu vấn đề (bug tracker)
* Hai chiến lược chính:
  + Hướng dẫn người dùng về cách xác định và báo cáo bug một cách chính xác.
  + Sử dụng "hệ thống bạn bè"

**IRC / REAL-TIME CHAT SYSTEMS (TRÒ CHUYỆN THỜI GIAN THỰC)**

1. Sử dụng IRC cho Giao Tiếp Thời Gian Thực



- Chọn Tên Kênh

- Quản Lý Nhiều Kênh

- Thiết Lập Chủ Đề Kênh

- Sử Dụng Trang Web Pastebin

**BOT**

Bot trong Kênh IRC:

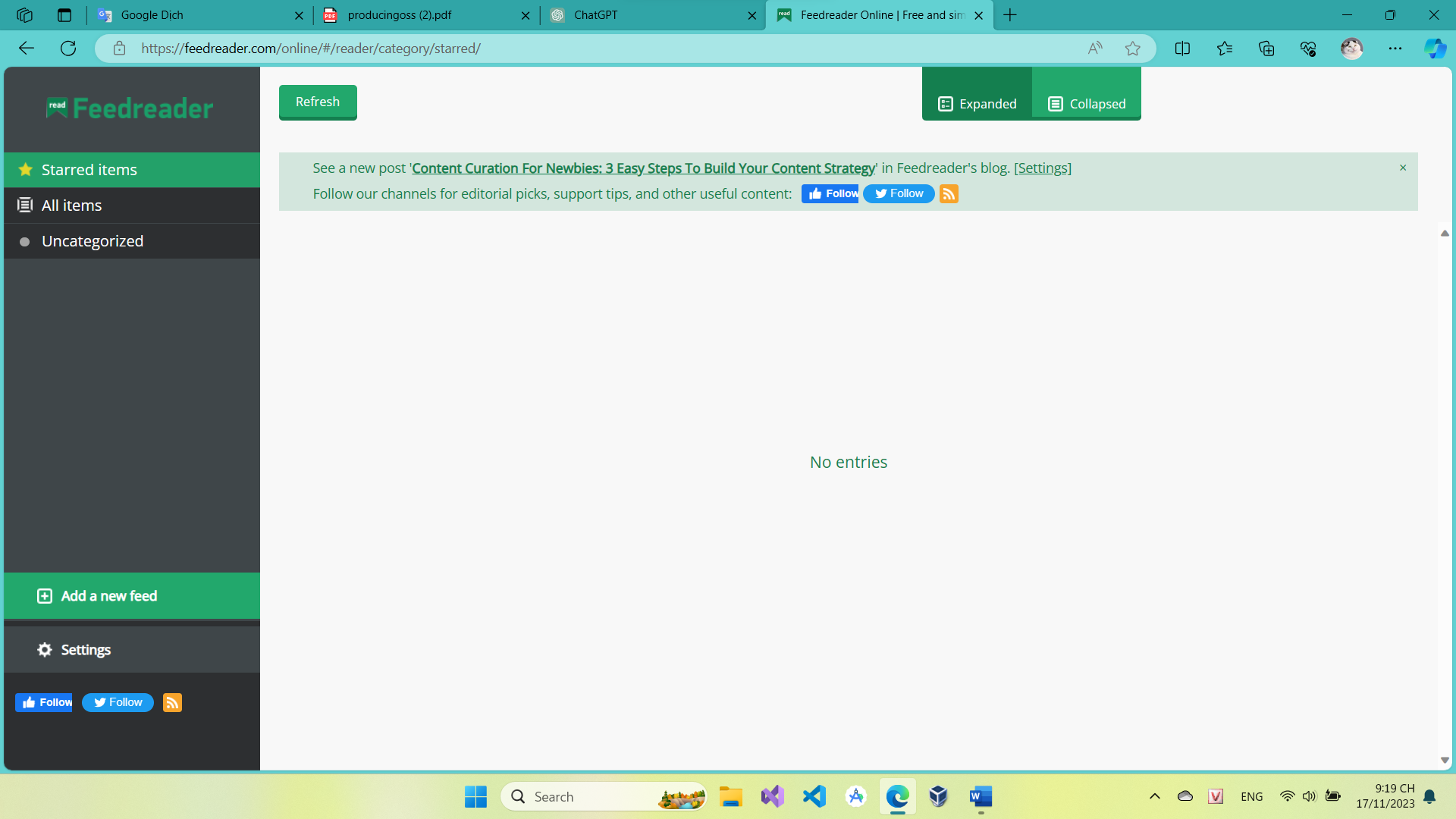
* Về việc sử dụng Bot
* Lợi Ích của Việc Sử Dụng Bot

**ARCHIVING IRC (LƯU TRỮ ĐOẠN CHAT)**

Mặc dù có khả năng lưu trữ mọi thứ, nhưng việc này không được mong đợi do cuộc trò chuyện trên IRC thường được xem là nửa riêng tư. Việc lưu trữ cần được thực hiện cẩn thận để tránh làm phiền người dùng. Nếu quyết định lưu trữ tất cả, cần thông báo rõ trong chủ đề kênh và cung cấp URL đến lưu trữ.

**RSS FEEDS**

RSS (Really Simple Syndication) đây là một công nghệ cung cấp một cơ chế tự động để cập nhật thông tin từ các trang web, blog, hoặc nguồn tin tức trực tuyến khác.



**WIKIS**

* Wiki là một trang web cho phép mọi người đọc chỉnh sửa hoặc mở rộng nội dung của nó
* Wiki chưa phải là tiêu chuẩn cho các dự án mã nguồn mở, nhưng có khả năng sẽ trở thành như vậy trong tương lai.
* Một số vấn đề thường gặp với như sự trùng lặp thông tin, và mục tiêu khán giả không nhất quán, thông tin không chính xác.

**WEB SITE**

Về việc thiết lập trang web cho dự án từ góc độ kỹ thuật. Nếu không có kiến thức, việc tìm kiếm sự giúp đỡ từ người có kinh nghiệm là một lựa chọn, nhưng để tiết kiệm thời gian và công sức, nhiều người chọn sử dụng các trang web hosting đã được chuẩn bị sẵn.

***CANNED HOSTING:***

Ưu điểm của Canned Hosting

- Công suất và băng thông máy chủ

- Sự đơn giản.

Nhược Điểm của Canned Hosting:

- Giới hạn sự lựa chọn.

- Kiểm soát hạn chế.

- Kiểm soát thiết kế: bạn có thể không thiết kế trang web như mong muốn, và giao diện mặc định của trang web hosting không linh hoạt.

***CHOOSING A CANNED HOSTING SITE:***

1.Các trang web hosting phổ biến

- SourceForge: Là trang web hosting lớn và nổi tiếng nhất. [SourceForge](http://www.sourceforge.net/)

- savannah.gnu.org: Cung cấp dịch vụ tương tự. [savannah.gnu.org](http://savannah.gnu.org/)

- BerliOS.de: Một trang web khác cung cấp các dịch vụ tương tự. [BerliOS.de](http://www.berlios.de/)

2.Cung cấp hosting miễn phí

- Apache Software Foundation:[Apache Software Foundation](http://www.apache.org/)

- Tigris.org: [Tigris.org](http://www.tigris.org/)

***ANONYMITY AND INVOLVEMENT (ẨN DANH VÀ SỰ THAM GIA)***

* Vấn đề của việc sử dụng chức năng đăng nhập người dùng trên các trang web hosting.
* Đề xuất lựa chọn cẩn thận về việc cho phép thực hiện một số hành động mà không yêu cầu đăng nhập.