



Introducción a los Repositorios de Código Distribuido

Evolución de Software

Diego Madariaga

Contenidos de la clase

1. Organización del servidor de Discord
2. Problemas en la evolución de software
3. Minería del historial de un repositorio

1.

Organización del servidor de Discord

Servidor de Discord

- ▷ Únanse al servidor (link en el chat)
- ▷ Dedicado principalmente a los laboratorios
- ▷ Lo usaremos en esta primera clase para organizarnos

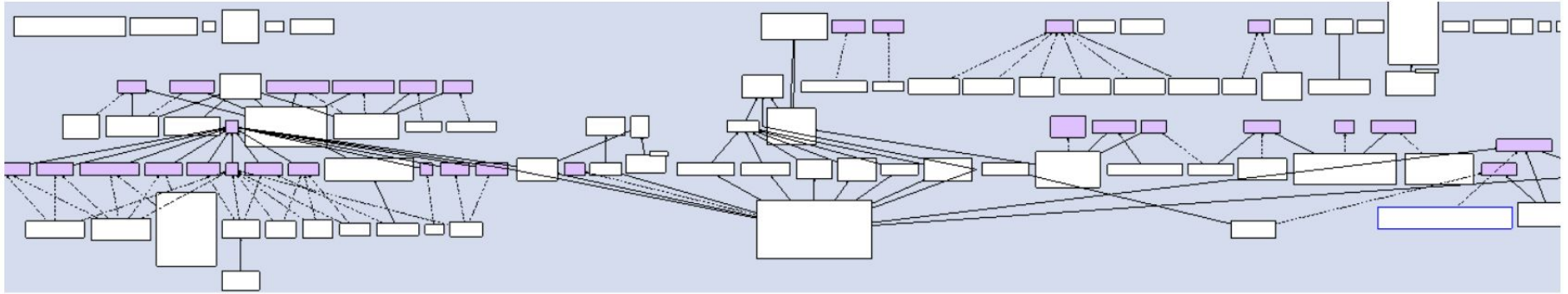
2.

Problemas en la evolución de software

Software en constante evolución

- ▷ Cambios en los requerimientos
- ▷ Implementación de nuevas funcionalidades
- ▷ Cambios en los desarrolladores
- ▷ Cambios en el ambiente

Software en constante evolución



Software en constante evolución

- ▷ Es necesario considerar la variable temporal para entender la evolución de un software
- ▷ Historial de cambios (sistema de control de versiones)

Historial de cambios

- ▷ Sistema de control de versiones:
 - Guarda registro de cada cambio (quién, qué, cuándo y por qué)
- ▷ Git: Historial de commits

Historial de cambios (git log)

```
$ git log --stat
commit ca82a6dff817ec66f44342007202690a93763949
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Mon Mar 17 21:52:11 2008 -0700

    Change version number

Rakefile | 2 +-
1 file changed, 1 insertion(+), 1 deletion(-)

commit 085bb3bcb608e1e8451d4b2432f8ecbe6306e7e7
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 16:40:33 2008 -0700

    Remove unnecessary test

lib/simplegit.rb | 5 -----
1 file changed, 5 deletions(-)

commit a11bef06a3f659402fe7563abf99ad00de2209e6
Author: Scott Chacon <schacon@gee-mail.com>
Date:   Sat Mar 15 10:31:28 2008 -0700

    Initial commit

README           | 6 ++++++
Rakefile         | 23 ++++++++++++++++++++++
lib/simplegit.rb | 25 ++++++++++++++++++++++
3 files changed, 54 insertions(+)
```

3.

Minería del historial de un repositorio

Minería del historial de un repositorio

- ▷ ¿Qué información del código se puede obtener analizando el historial de cambios?
- ▷ ¿Qué información de los desarrolladores se puede obtener analizando el historial de cambios?

*Piense en información que pueda servir a futuro

Identificación de Hotspots

- ▷ Hotspot: Sección de código de interés
 - Mezcla de esfuerzo + complejidad
 - Frecuencia de cambios + cantidad de líneas afectadas
- ▷ Partes del código que requieren más atención
 - Verificación de funcionamiento
 - Actualización de diseño

Perfil de desarrolladores

- ▷ Identificación de equipos
- ▷ Modo de desarrollo
- ▷ Expertiz de cada desarrollador



Introducción a los Repositorios de Código Distribuido

Evolución de Software

Diego Madariaga