Git版控起手式

OAD-WebTeam / Dave Mark

議程

- Git & Gitea 版控
- CI/CD 自動佈署
- 參考資料

Git & Gitea 版控

“ https://git-scm.com/ ”

Git vs SVN

	Git	SVN
管理概念	分散式(distributed) 強調個體	集中式(centralized) 中央伺服器儲存
效能	fast local commit push server	slow commit to server
離線工作	○	✕
分支掌控度	high 清楚知道Merge版本歷程	low 無法分辨Merge版本
適用多人開發	較適合(搭配Flow)	較困難
系統檔案	根目錄.git folder & .gitignore File	每層目錄放.svn folder
權限	使用帳號角色劃分	針對目錄做存取權限

Git Commit Message

- feat：新增或修改功能
- fix：修補 bug
- docs：文件
- style：格式
- refactor：重構
- perf：改善效能
- test：增加測試
- chore：maintain
- revert：撤銷回覆先前的 commit

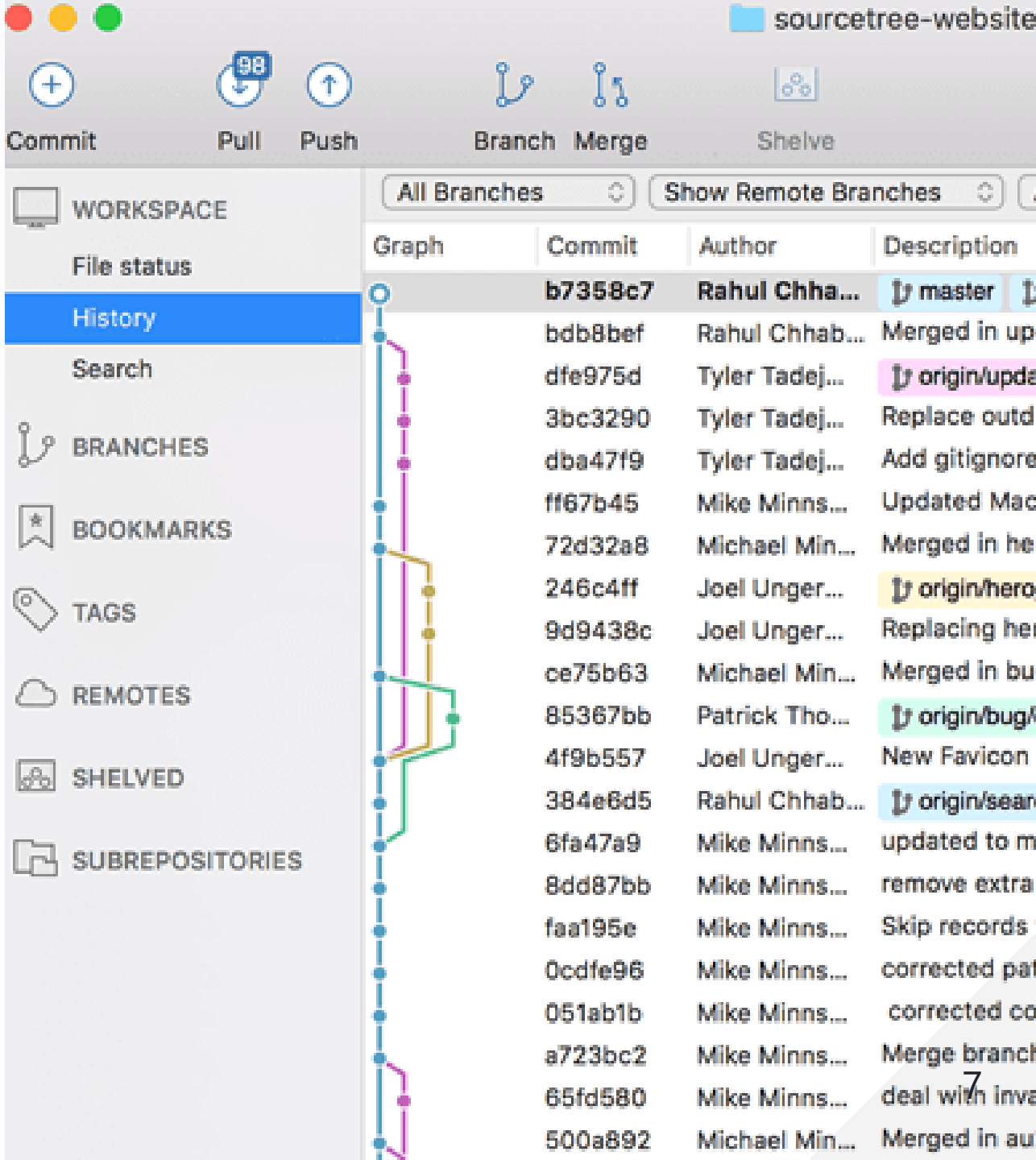
Tools

Sourcetree	Visual Studio Code	Visual Studio
		

SourceTree

“ https://www.sourcetreeapp.com/ ”

OAD-WebTeam / Dave Mark



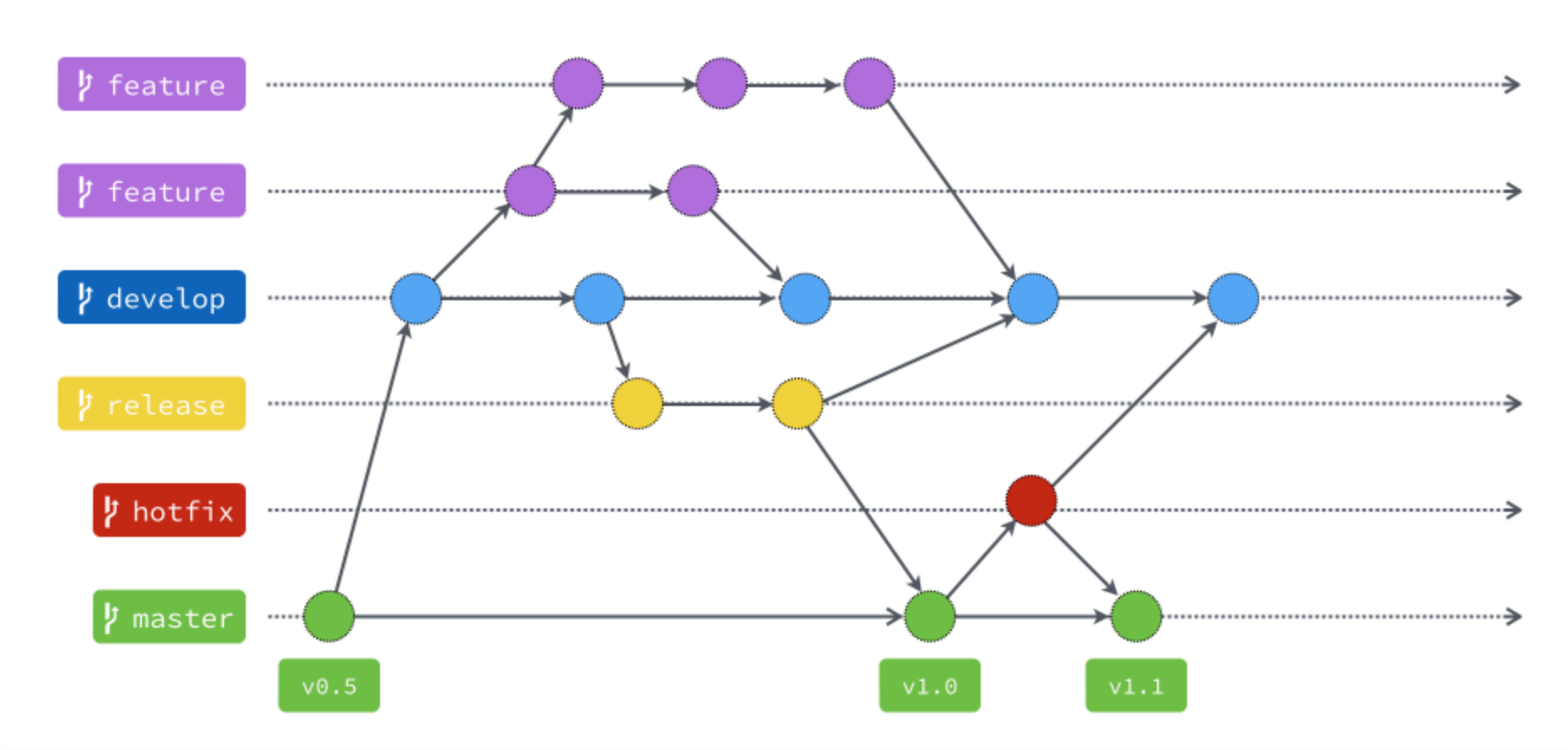
“ https://about.gitea.com/ ”

Gitea

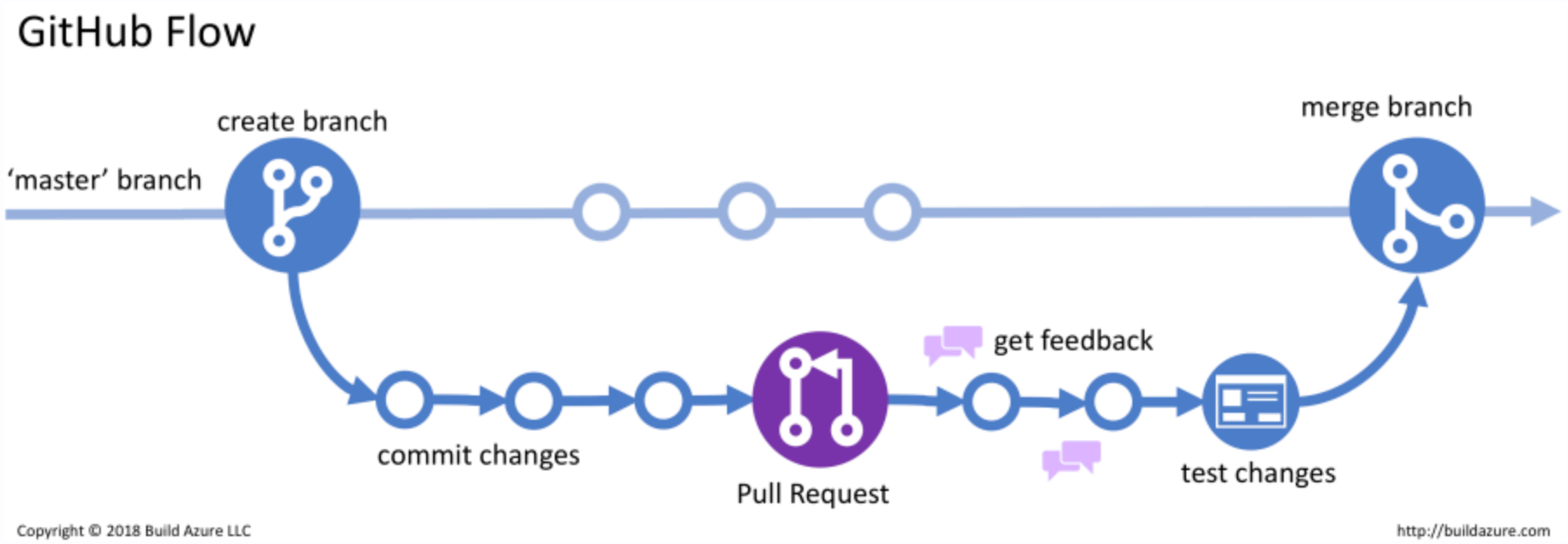
- 跨平台：使用 **Go** 語言開發，可在 Windows、macOS、Linux 執行。
- 安裝：
  - Windows 搭配 Windows 服務
  - Docker image 建置 Linux 服務
- 運作方式：比照 GitHub 方式以個人與組織方式運作，各 Repo 存取權限依照定義的帳號**角色**呈現。
- CI/CD：搭配 **Gitea Action** 實作

Git Flow vs Github Flow vs GitLab Flow ?

Git Flow



GitHub Flow


GitHub Flow
create branch
merge branch
'master' branch
get feedback
commit changes
Pull Request
test changes
Copyright © 2018 Build Azure LLC
http://buildazure.com

實戰演練 Demo

“Tutorial_Git-UploadOldProject”

CI/CD 自動佈署

什麼是 CI/CD ?

- 持續整合(Continuous Integration, CI)
  - 建置：確保提交的程式碼是否可執行
  - 測試：確保功能正常與軟體品質
  - 程式碼分析：檢查 code style 或程式的穩健度
- 持續交付/佈署(Continuous Delivery / Deployment, CD)
  - 持續交付：自動將生產已就緒版本發佈到代碼存儲庫
  - 持續佈署：可以自動將應用發佈到生產環境

CI/CD 準備工作

- 持續整合(Continuous Integration, CI)
  1. 專案以純 Source Code 為主
  2. 流程是否已可使用 command 處理?
  3. 建置環境是否備妥? SDK?
- 持續交付/佈署(Continuous Delivery / Deployment, CD)
  1. 流程是否已使用 command 處理?
  2. 目的與 Gitea Server Protocol 是否互通? 認證機制?

Gitea workflow

- Gitea 專案開啟 `Actions` 功能
- 專案內建置 `.gitea/workflow/{腳本名稱}.yml`
- 觸發條件為何? `push branch develop` or `push branch release`?
- 工作事項簡單區分 build(CI) 及 deploy(CD) job
- 撰寫 command 腳本 (ex. Command Prompt、Windows PowerShell)
- 設計 Secrets 參數 (ex. 主機資訊、認證)
- Action Runner 工作事項

實戰演練 Demo

參考資料

- Gitbook
- Git Commit
- Git Flow