

Computadors

Fonaments de la Programació

(1/2)

Grau en Ciència i Enginyeria de Dades

Facultat d'Informàtica de Barcelona (FIB)

Universitat Politècnica de Catalunya (UPC)

Creative Commons License

Aquest document utilitza Creative Commons Attribution 3.0 Unported License



Els detalls d'aquesta licència es poden trobar a <https://creativecommons.org/licenses/by-nc-nd/4.0>

Index

- **Gestió de Projectes de Software**
- Programació i Compilació

Index











- Entendre la teva aplicació
- Desenvolupament d'un Projecte
- Control de Versions
- Compilació, Depuració i Eines de Gestió de Codi
- L'Importància de fer Tests
- Distribució de Codi Automatitzat

Entendre la Teva Aplicació

- Objectiu de l'aplicació
- Algorismes
- Estructures de Dades
- Quins serveis usa del sistema
- Estructura del Codi
 - Directoris
 - Fitxers capçalera
 - Fitxers de codi font
- Estructura dels fitxers binaris
 - Quan s'han de compilar...

Entendre la Teva Aplicació

- Exemple...  Geany git

 data	Theme improvements (#1382)
 doc	manual: added documentation about replacement of 'untitled.ext' with ...
 icons	icon: regenerate png/ico files based on the svg
 m4	Update Scintilla to version 3.7.5 (#1503)
 plugins	filebrowser: Don't change directory on project save
 po	Small update of German translation
 scintilla	Update Scintilla to version 3.7.5 (#1503)
 scripts	Update Scintilla to version 3.7.5 (#1503)
 src	Merge pull request #1748 from kugel-/msgwin-api
 tests	bash may not found in the system (#1574)



Entendre la Teva Aplicació

- autotools
- src
- include
- libs

 Makefile.am	Merge pull request #1095 from eht16/issue1076_win32_build_working_dir...
 about.c	Remove a space (#1790)
 about.h	Normalize use of header guards and extern "C" guards
 app.h	Add utils_get_real_path() and use it
 build.c	Work around a `-Wformat-overflow` warning
 build.h	doxygen: various doxygen-related fixes in preparation for gtkdoc gene...
 callbacks.c	Show status message on attempt to execute empty context action.
 callbacks.h	Allow to set a keybinding for File->Properties
 dialogs.c	Fix canceling keybinding overriding by discarding the dialog
 dialogs.h	Protect private definitions by the GEANY_PRIVATE macro in headers
 document.c	Add missing space in string. Fixes #1789
 document.h	Added option to auto reload files changed on disk (#1246)
 documentprivate.h	Add support for Keyed Data Lists for documents
 editor.c	Remove some unused variables

Suport de Biblioteques segons Llenguatge i SO

python: **libpython3.6m.so**

libpthread.so

libdl.so

libm.so, libmvec.so

libc.so

xclock: **libX11.so**

libXext.so

libm.so

libc.so

...

cpmd.x.omp: **libmpi.so**

libgfortran.so

libpthread.so, **libgomp.so**

libm.so, libmvec.so

libc.so

...

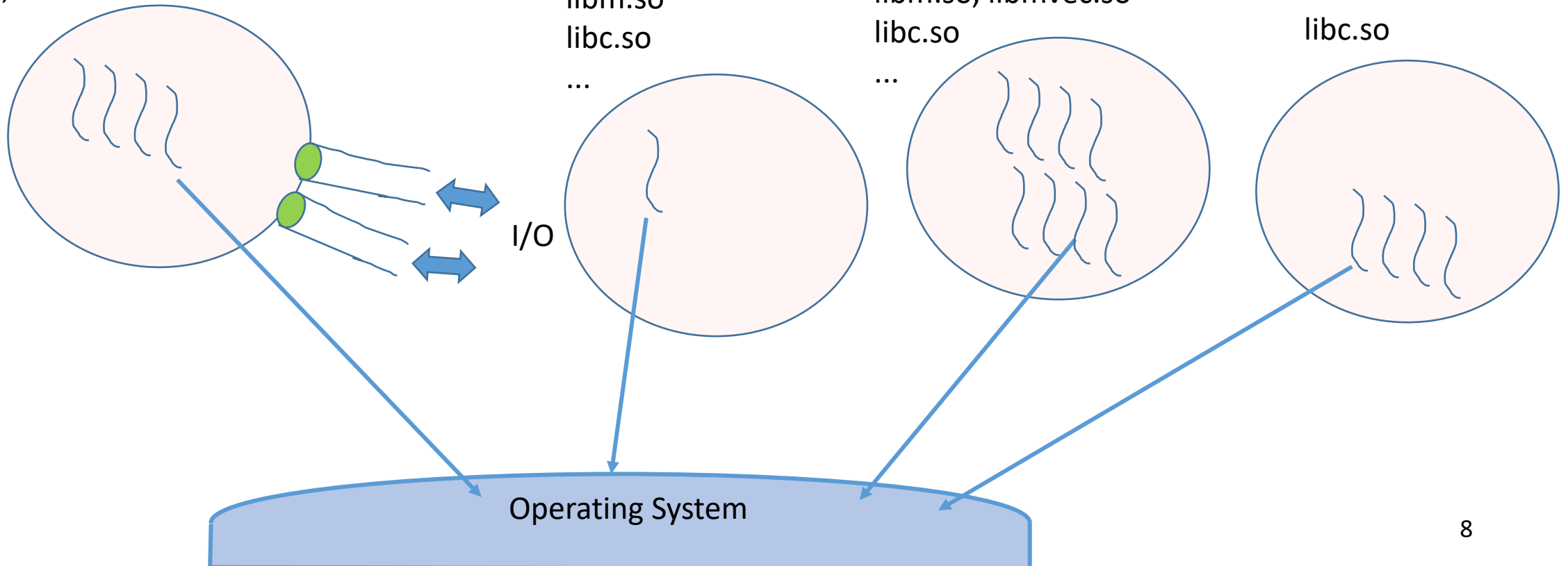
perl: **libdl.so**

libcrypt.so

libpthread.so

libm.so

libc.so

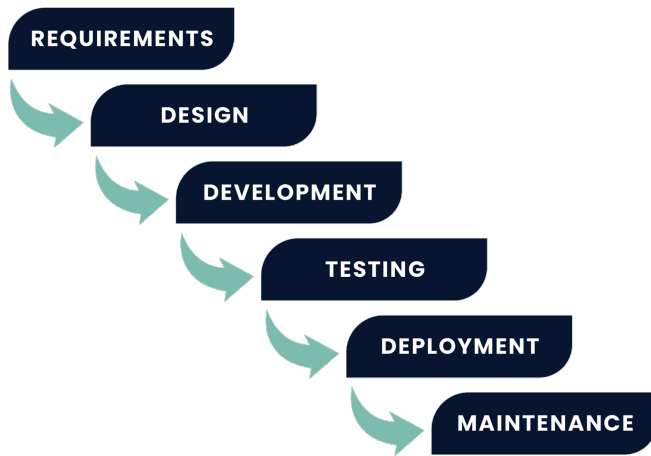


Metodologies de Desenvolupament

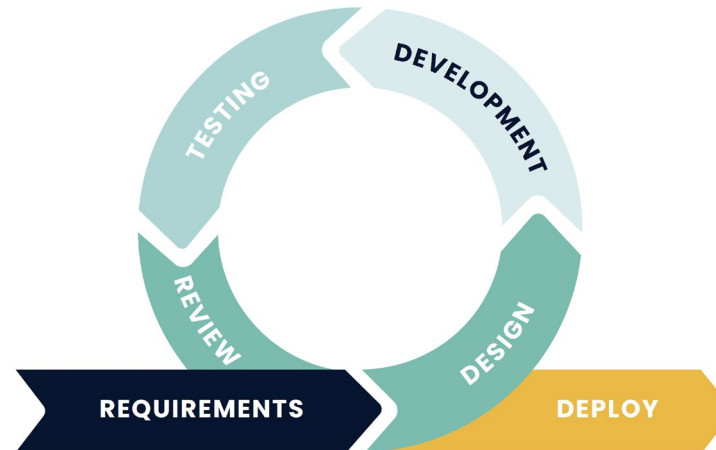
- Cascada (Waterfall)
 - Treball desgloçat en fases seqüencials
 - Requeriments->Diseny->Desenvolupament->Testing->Desplegament->Resultat Final
 - Quan el temps/costos són limitats, i els requisits/abast del producte són clars des del principi
- Àgil (Agile)
 - Equip multi-funcional que va descobrir i construir mitjançant un procés iteratiu
 - Mètode Scrum: treball desgloçat en iteracions conegudes com sprints
 - Quan els detalls no són clars des del principi i els requisits/abast es van descobrir/definint/adaptant mitjançant diferents iteracions (es van fent micro entregues)
- Model Híbrid

Metodologies de Desenvolupament

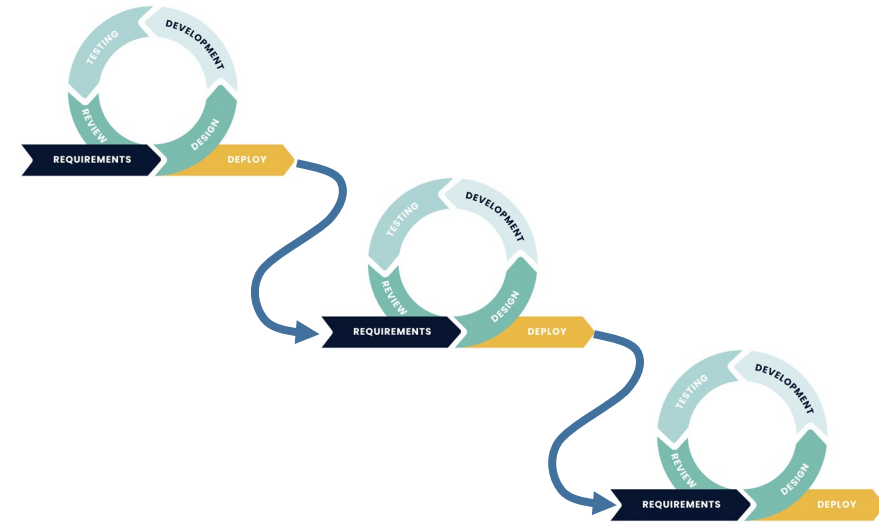
Cascada



Àgil



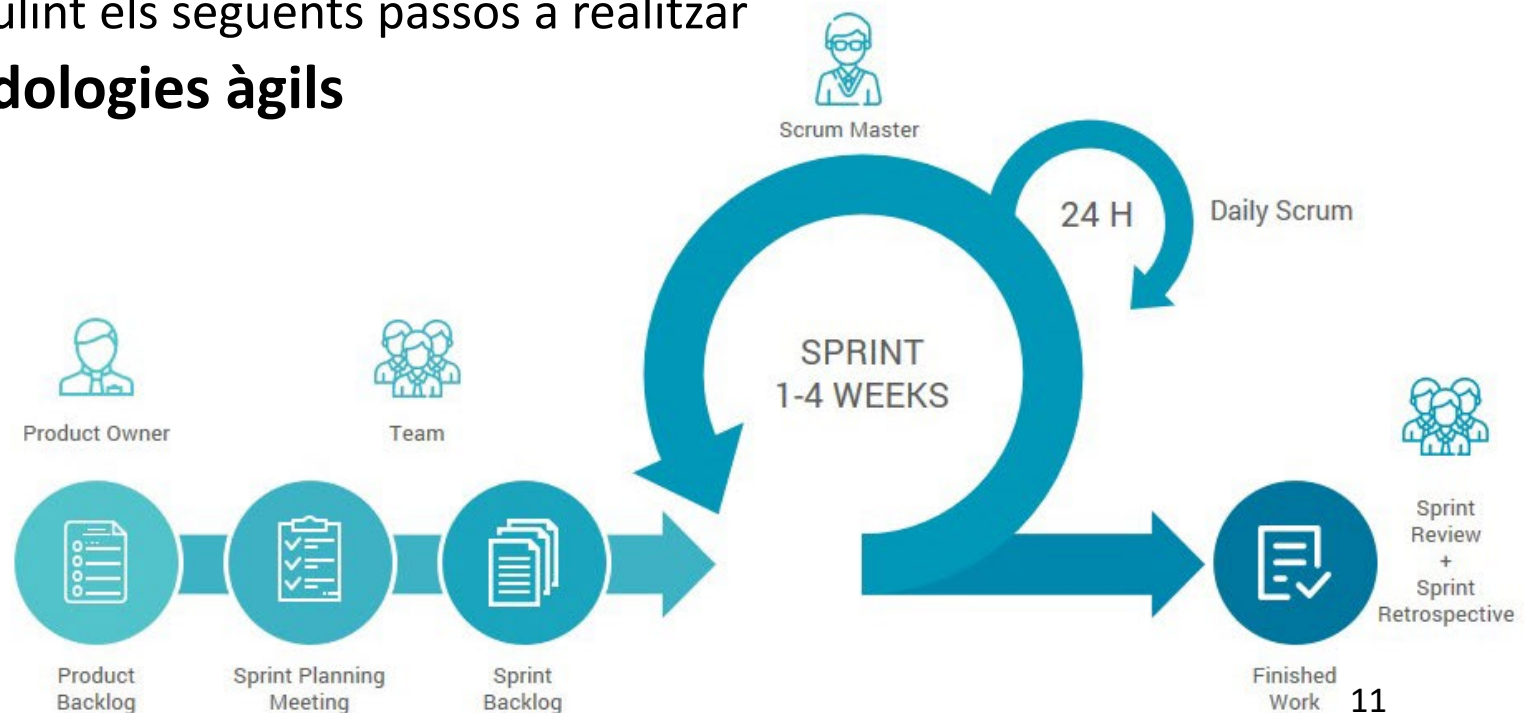
Híbrid



Imatges del llibre: [The Perfect Blend: How to use Jira for waterfall and hybrid projects](#) (C. Berry, Nov 2021)

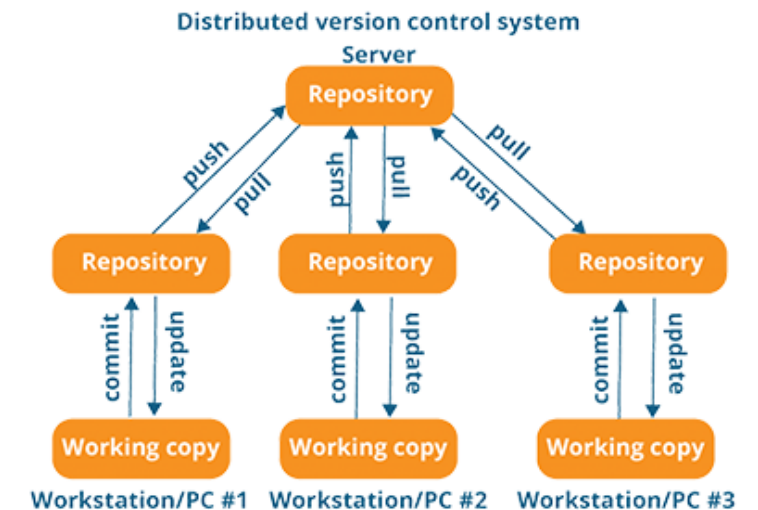
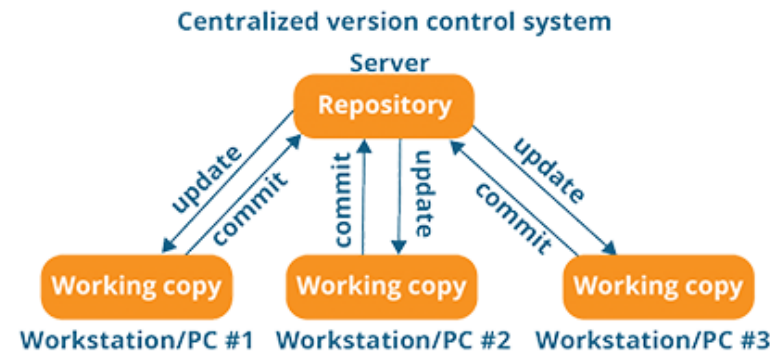
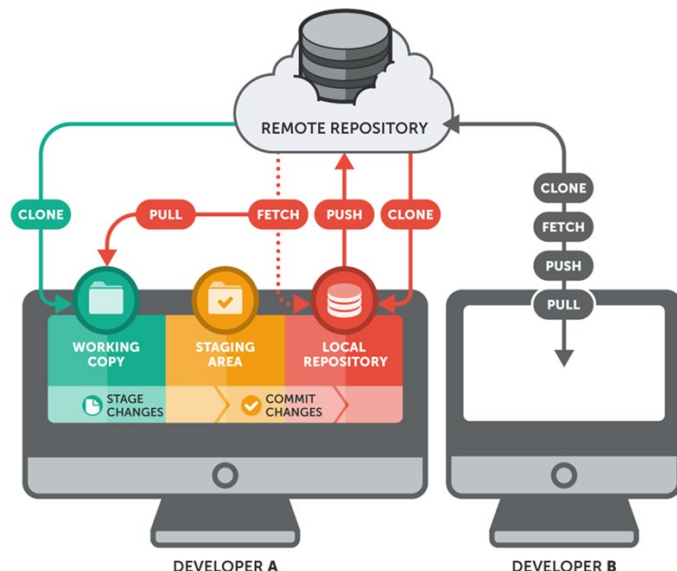
SCRUM

- SCRUM és un marc de treball per la gestió de projectes
 - Centrat en la millora continua
 - Parteix del fet que l'equip no sap tot el que s'ha de fer des del principi
 - En cada iteració es van pulint els següents passos a realitzar
 - **Molt apropiat per metodologies àgils**
- Ben estructurat
 - Diversos Rols
 - Varies Etapes



Gestió del Versionat de Codi

- Eines de control de versions
 - Un conjunt d'eines que ajuda a mantenir un registre dels canvis en el codi mitjançant una jerarquia de estructures internes i fitxers que ajuden a gestionar diferents versions concurrents de codi
 - Centralized Version Control System (e.g. cvs, svn): hi ha una **única còpia del repositori**
 - Distributed Version Control System (e.g. **git**, Mercurial): hi ha **múltiples còpies del repositori**
 - Hi ha repositoris accessibles via web per facilitar la gestió de codi
 - E.g.: Github, Gitlab



Conceptes Bàsics (exemples basats en Git)

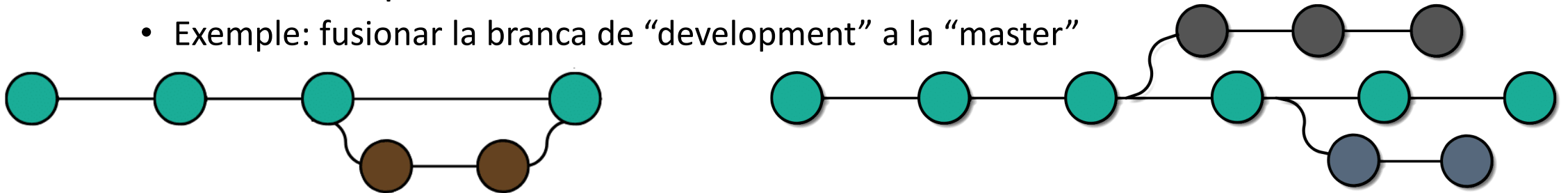
- Repositori
 - L'enregistrament de tots els canvis en els docs d'un Projecte
 - Carpeta ".git"
 - **Clone**
 - Còpia un repositori existent a una carpeta local
- Flux de treball bàsic d'un entorn git

Director de Treball → Àrea de Preparació (Staging) → Repositori Git

 - **Add**
 - Afegeix un canvi des del directori de treball a l'àrea de preparació
 - **Commit**
 - Captura un canvi que has preparat prèviament en l'àrea de preparació
 - Detalls addicionals (e.g. timestamp)
 - **Push**
 - Actualitza el repositori remot: "puja" els canvis (commits)
 - **Pull**
 - Actualitza el repositori local: "descarrega" els canvis (commits)

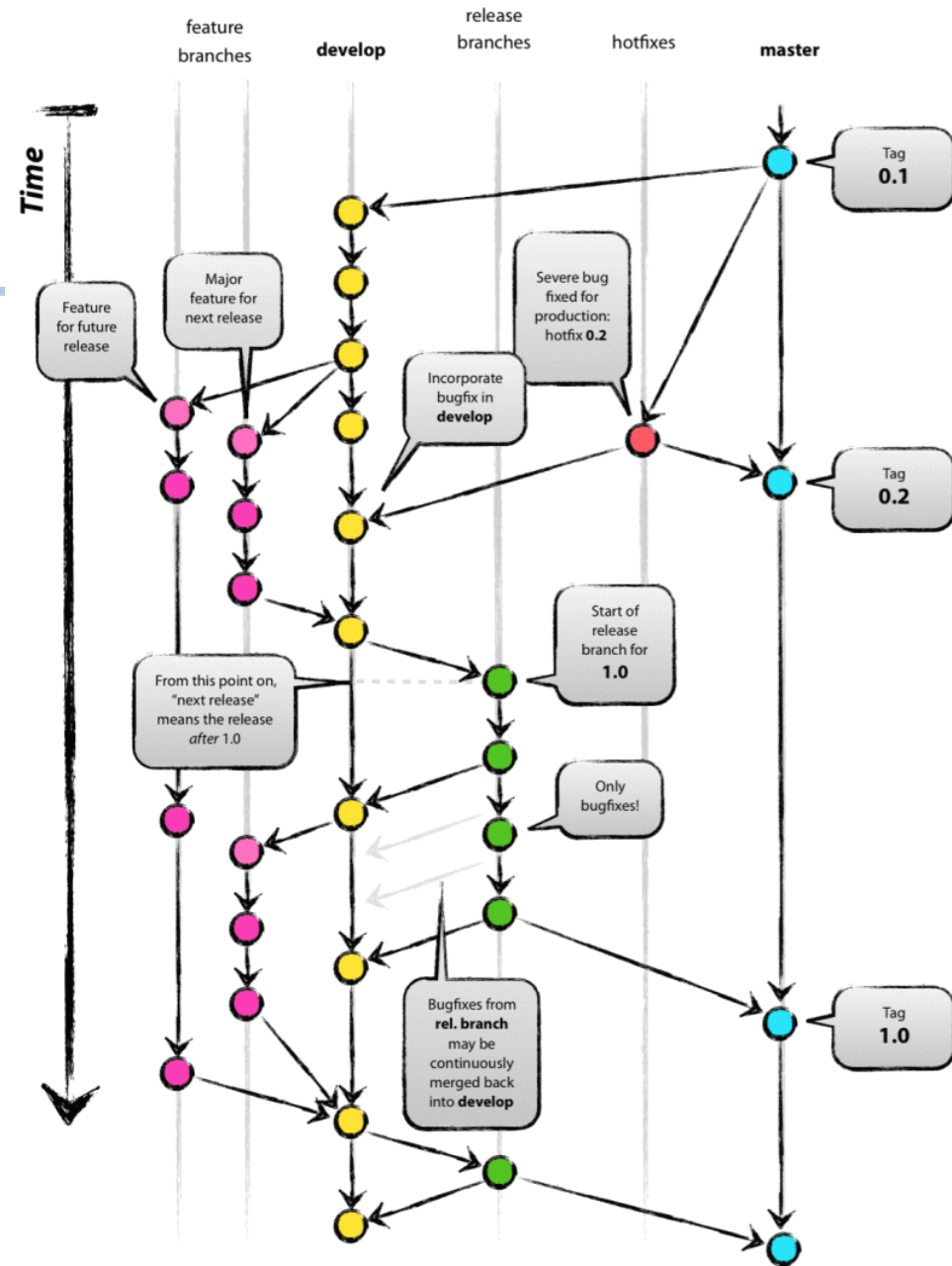
Conceptes Bàsics (exemples basat en Git)

- Branch
 - És una línia de treball per desenvolupar en un Projecte
 - Les modificacions en una branca no afecten a altres
 - Com a mínim hi ha una branca en un repo: la **master**
 - És la branca principal
 - Gestió de branques
 - Exemple: fusionar la branca de “development” a la “master”



- Pots canviar entre branques i entre commits

Ex. Gestió de Codi



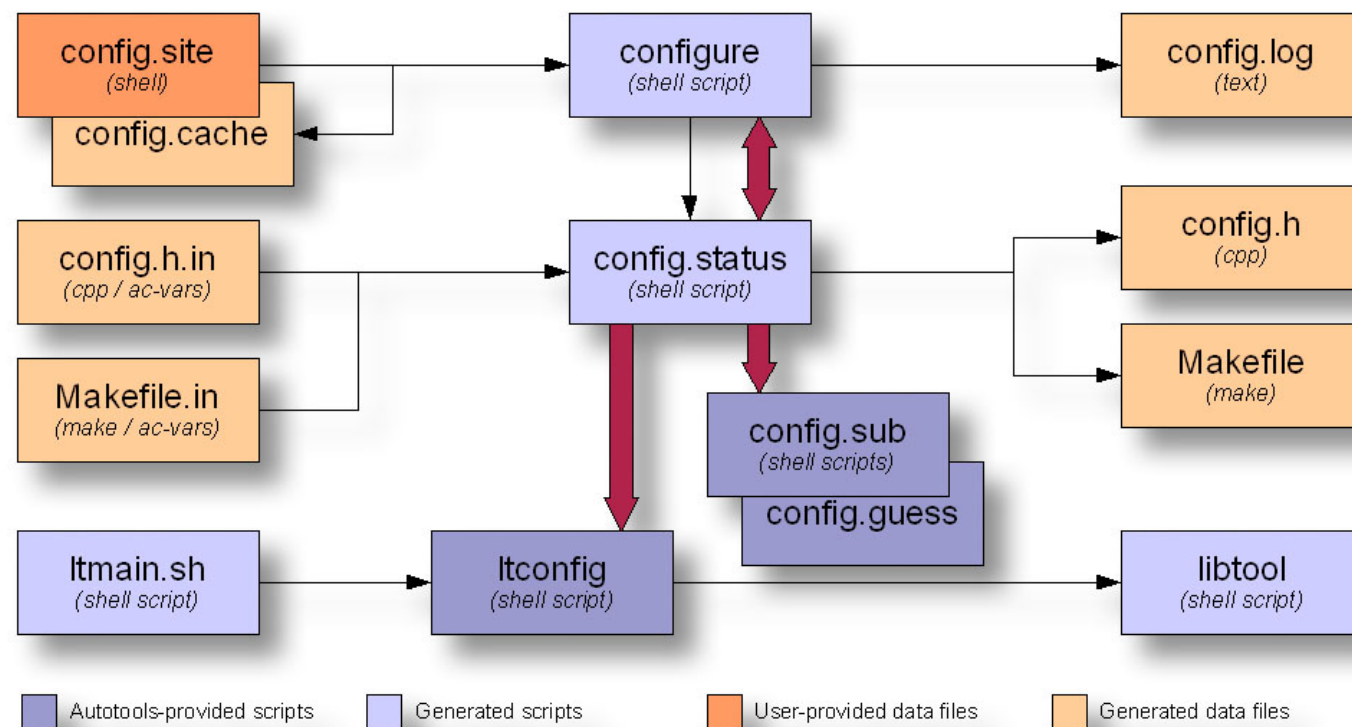
Compilació i Eines de Gestió de Codi

- Formats de Versions de Software
 - (X.Y.Z)
 - X: Canvis grans (major changes), normalment incompatibles amb versions anteriors
 - Y: Canvis menors (minor changes), noves funcionalitats compatibles amb versions anteriors
 - Z: Revisions/Parches per solucionar errors/incidències (bug fixing)
 - Sistema de nombres senars-parells
 - Nombres senars per versions en desenvolupament i parells per versions estables
- Eines per fer el procés de construcció de l'aplicació (Build Process)
 - Un conjunt d'eines per desenvolupadors de software per crear/distribuir automàticament codi font i crear paquets portables de software
 - L'eina autotools (de GNU) facilita donar suport a la portabilitat donant eines per automatitzar la configuració i creació del procés de compilació de tot un projecte software, basat en unes convencions estàndard, com ara noms estàndard de carpetes per saber on estan fitxers
 - autotools = autoconf + automake + libtool + ...
 - Cmake és la següent generació de l'eina autotools

Exemple d'un procés de compilació i instal·lació

(En el terminal...)

```
#> Download source code
...
#> ./configure
```

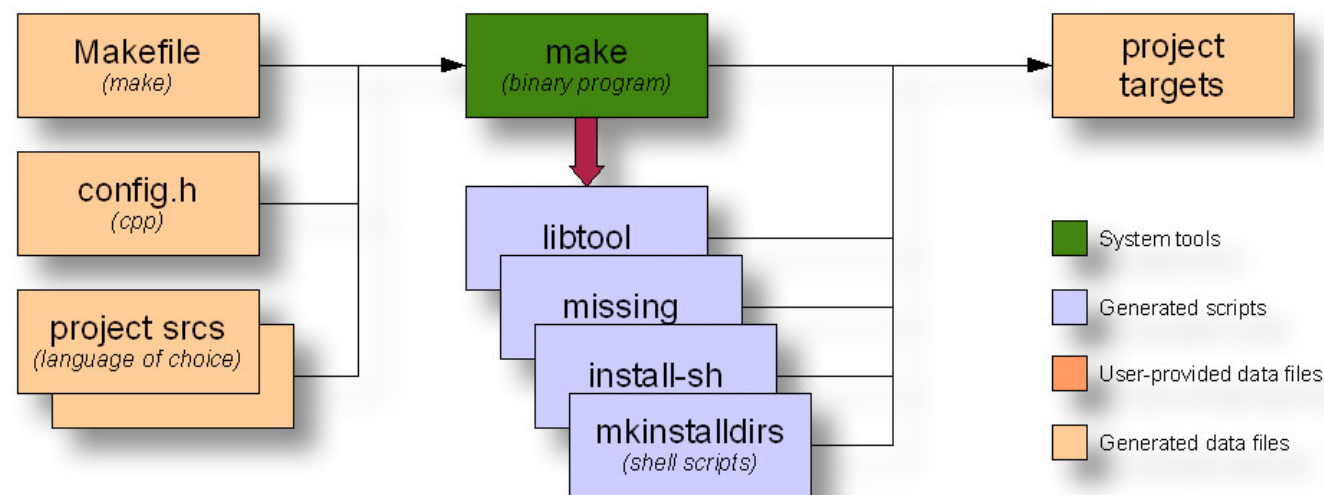


Procés de configuració: diagrama del flux de dades

Exemple d'un procés de compilació i instal·lació

(En el terminal...)

```
#> Download source code
...
#> ./configure
#> make all
#> make install
```



Procés de compilació i instal·lació: diagrama del flux de dades

Index

- Entendre la teva aplicació
- Desenvolupament d'un Projecte
- Control de Versions
- Compilació, Depuració i Eines de Gestió de Codi
- **L'Importància de fer Tests**
- Distribució de Codi Automatitzat

Tipus de Testing

- Des de aïllats a integrats i des de més ràpids a més lents:
 - **Test Unitari**: un unitat senzilla de codi (e.g. una nova funció)
 - **Test d'Integració**: dos o més unitats comunicades entre elles
 - **Test Funcional**: una característica
 - **Test d'Acceptació de l'usuari**: una característica, però des del punt de vista de l'usuari
 - **Test de Fum (Smoke Test)**: comprovar si el sistema és operacional
 - **Test de Regressió**: altres funcionalitats que ja estaven prèviament implementades
 - **Test d'Usabilitat**: relacionat amb l'eXperiència d'Usuari (UX)
 - **Test Exploratori**: la persona que fa els tests descobreix per si mateix les característiques i comprova tot el sistema
- Automatització de Tests: implementar tests automàtics

Desplegament Automàtic de Software

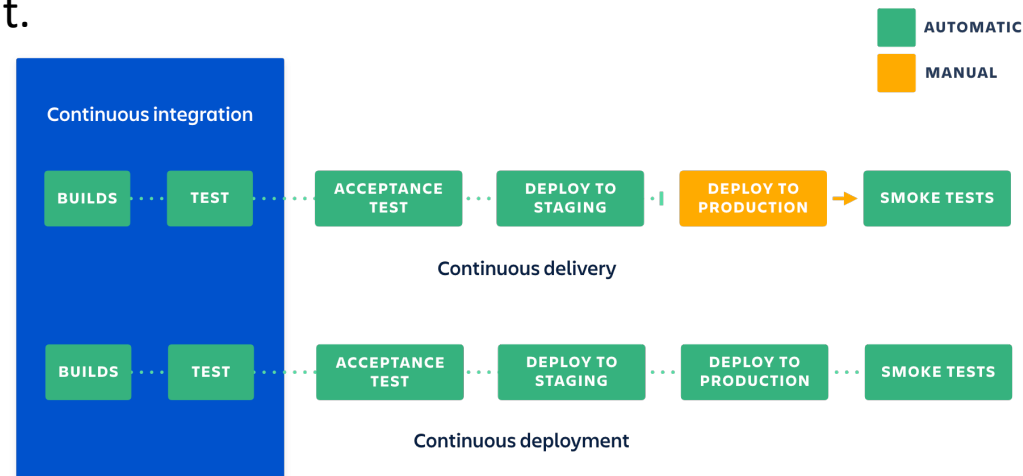
- Cada vegada que hi ha nous desenvolupaments, el Projecte s'ha d'actualitzar en una manera segura i per tant s'ha de testejar
 - Hi ha projectes de software que necessiten automatitzar aquest procés



- Hi ha diferents procediments depenent de l'objectiu
 - **Continuous Integration**
 - **Continuous Delivery**
 - **Continuous Deployment**
- Hi ha eines que integren varis eines per fer aquestes tasques
 - E.g. Jenkins, Ansible

Desplegament Automàtic de Software

- Hi ha diferents procediments depenent de l'objectiu
 - **Continuous Integration:** automatitza l'incorporació de nous desenvolupaments de codi. El principal objectiu és l'integració de codi a la branca principal de codi. Per tant, s'ha de testejar abans i després de l'integració
 - **Continuous Delivery:** automatitza la creació d'una nova versió de software. L'objectiu és tenir una versió nova que pot sortir com producte. Aquí són clau els tests d'acceptació d'usuari
 - **Continuous Deployment:** és un pas més, ja que desplega els resultats del Continuous Delivery, és a dir una nova versió, en un entorn, com ara producció. S'ha assumeix que tots els processos de tests s'han de completar amb èxit.



Bibliografia

- Git
 - <https://git-scm.com/doc>
- Gestió de Projectes (Agile + SCRUM)
 - <https://www.atlassian.com/agile>
- Introduction to software testing
 - https://en.wikipedia.org/wiki/Software_testing
- Continuous management of software releases
 - Atlassian Wiki
 - <https://www.atlassian.com/continuous-delivery/principles/continuous-integration-vs-delivery-vs-deployment>