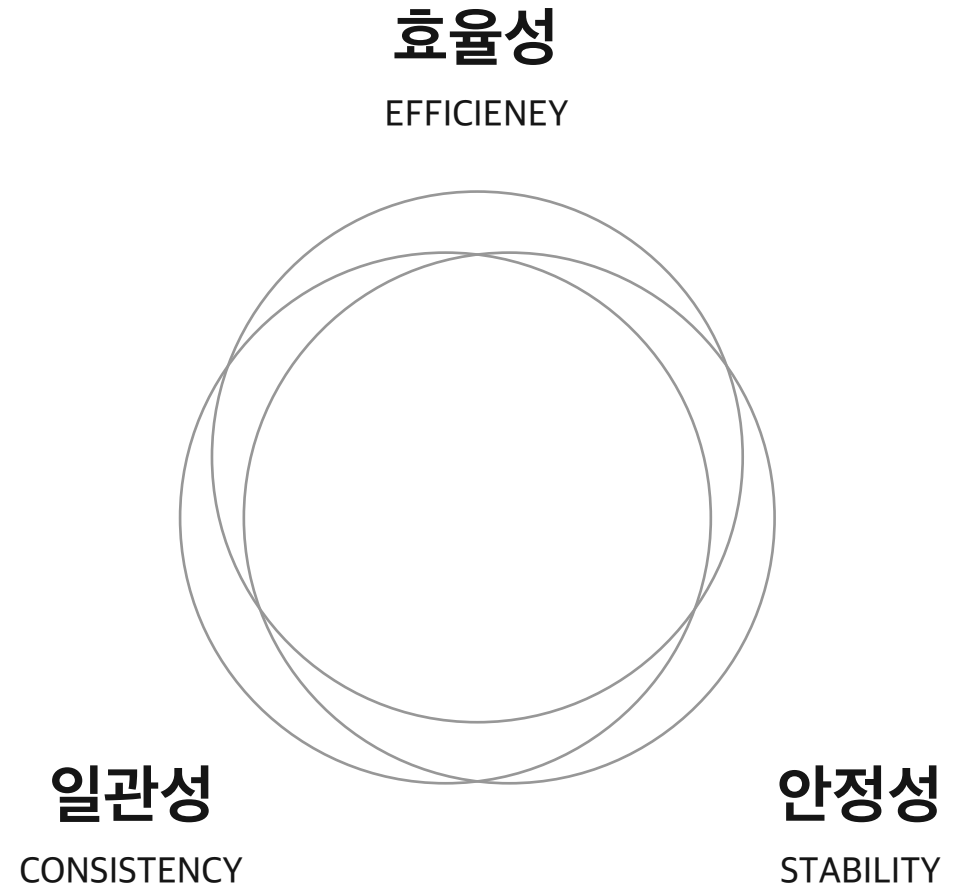


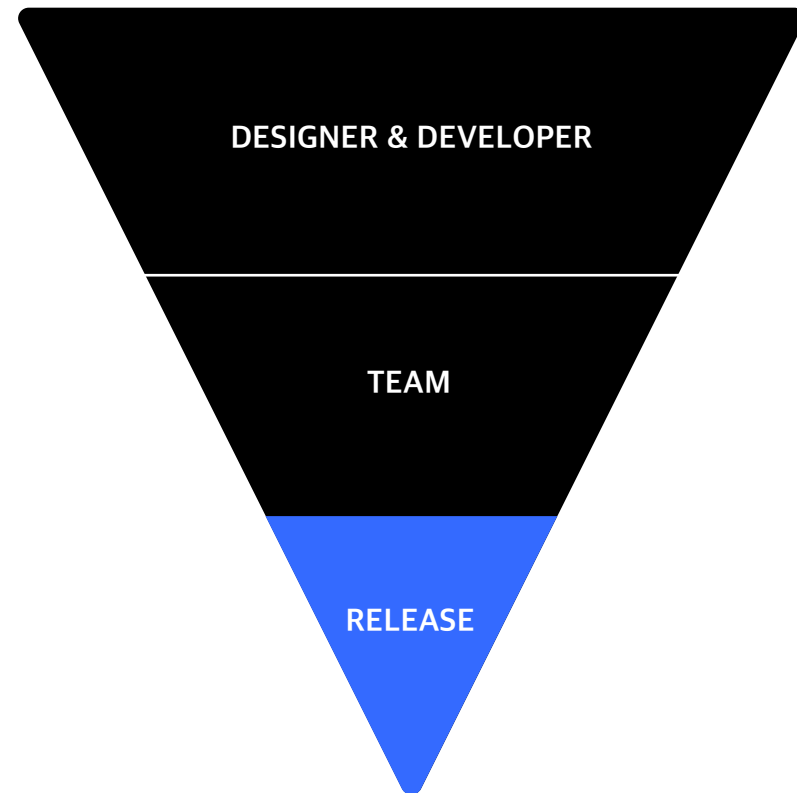
DESIGN VERSION CONTROL

Why need Versioning?

[= SUSTAINABLE DESIGN]



How to Versioning?



Funnel Structure

What is git or gitlab?

One Source, Multi Use

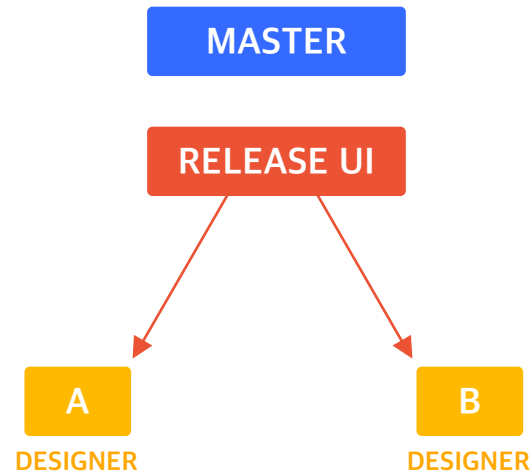
일관성

CONSISTENCY

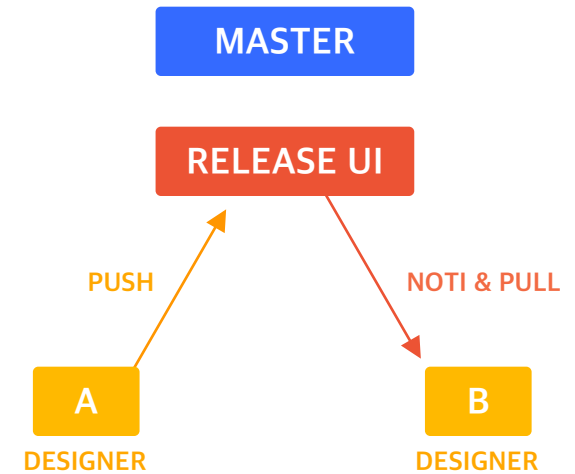
모든 디자이너가 공통의 최신 UI 소스 사용.

ASIS

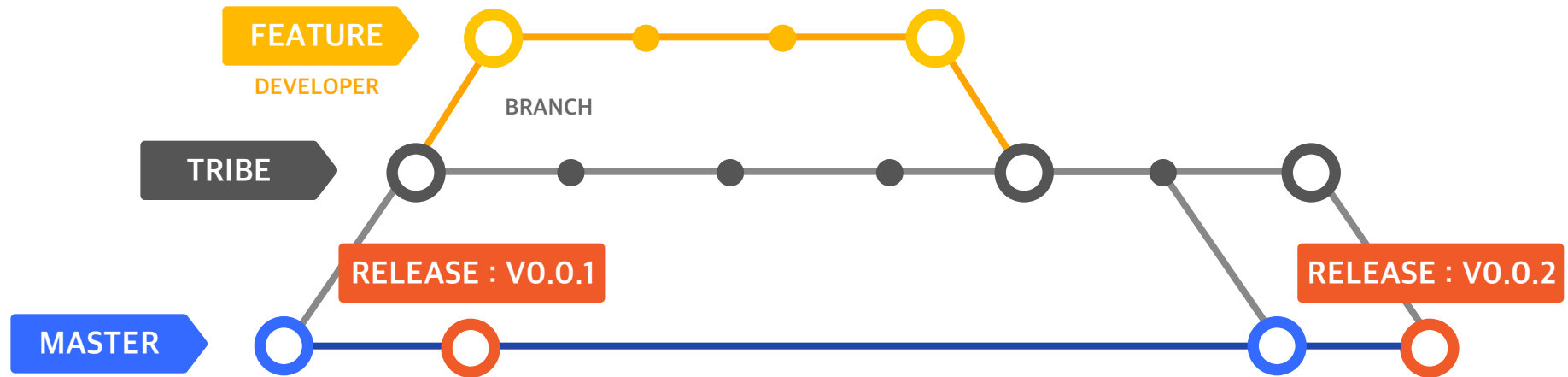
1. 최신 UI 소스의 담당자 확인 어려움.
2. 소스의 수정 여부 확인 어려움.



Design Component Download
: Common

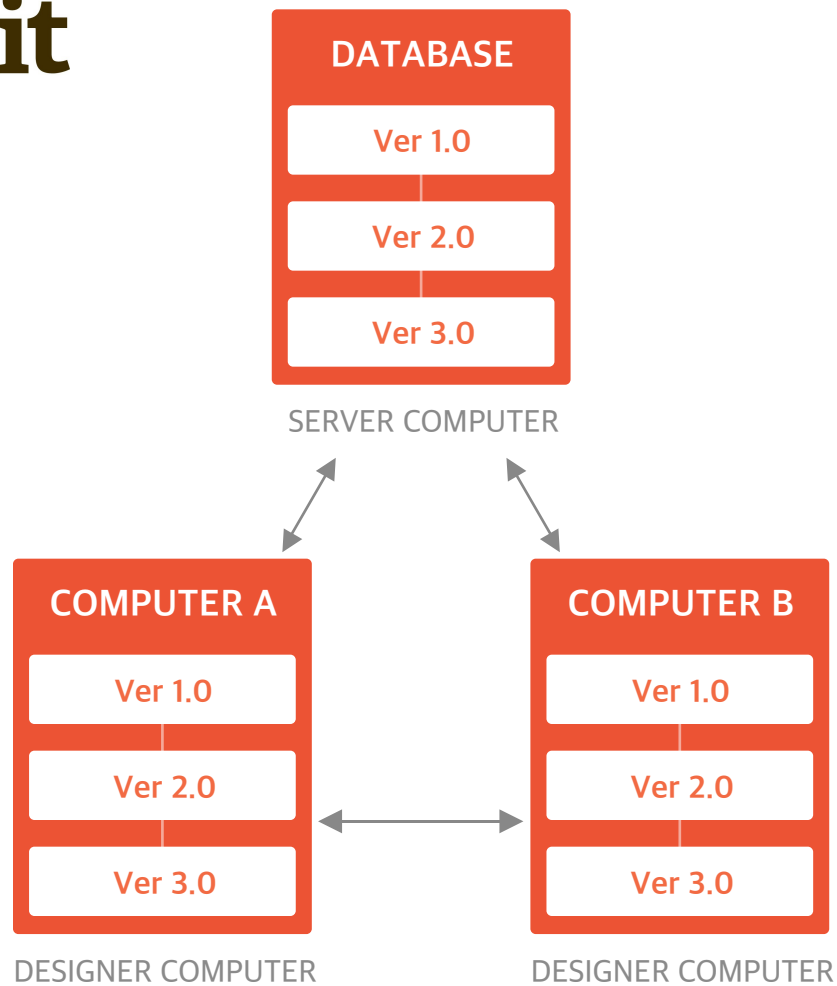


Design Update
: Latest



Developer Version Control System = Git

깃을 통해 모든 개발자의 소스코드의 버전이 관리된다.



깃 [: git]

버전관리 시스템의 중 하나로, 저장소를
통째로 복제하는 분산형 버전관리 시스템

버전관리 시스템의 종류

- 로컬 버전 관리 시스템
- 중앙 집중식 버전관리 시스템
- 분산형 버전 관리 시스템

<http://bit.ly/2tIOOQd>

Thank You

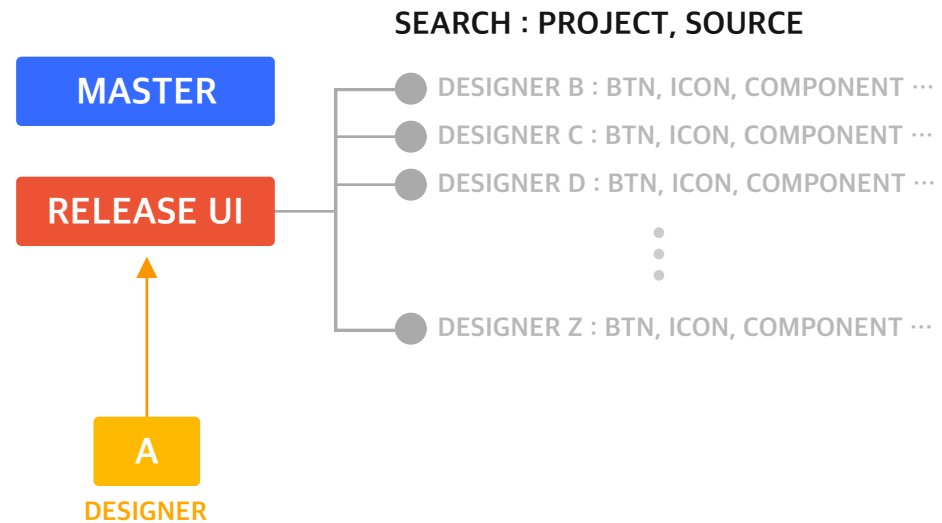
효율성

EFFICIENCY

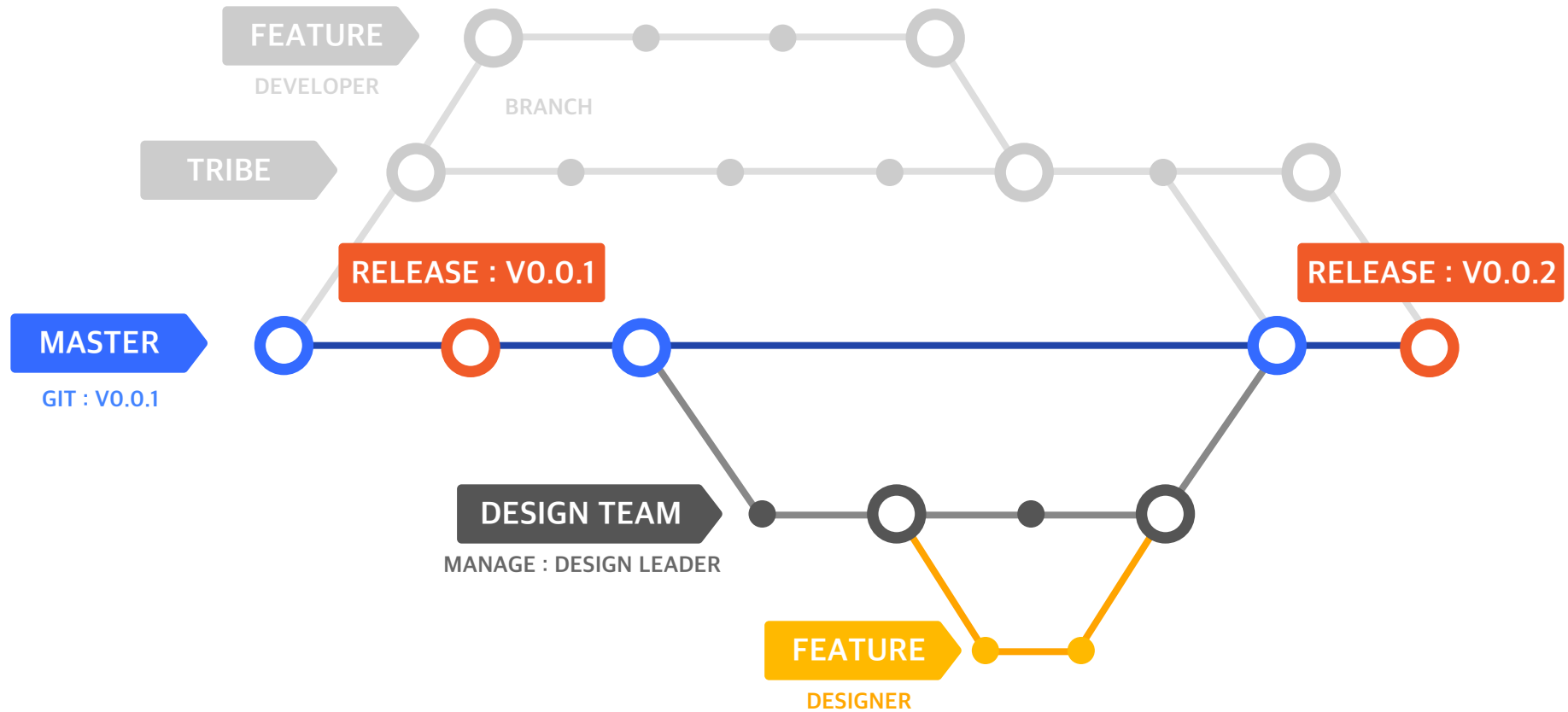
프로젝트 별 디자인 화면 또는 기능 즉시 검색 가능.
UI 히스토리 브라우징 가능.

ASIS

1. 프로젝트 별 화면 공유 안됨.
2. 유사/동일 기능 UI 반복 생성(이미지 소스 반복 생성)
3. UI 히스토리 확인 어려움.

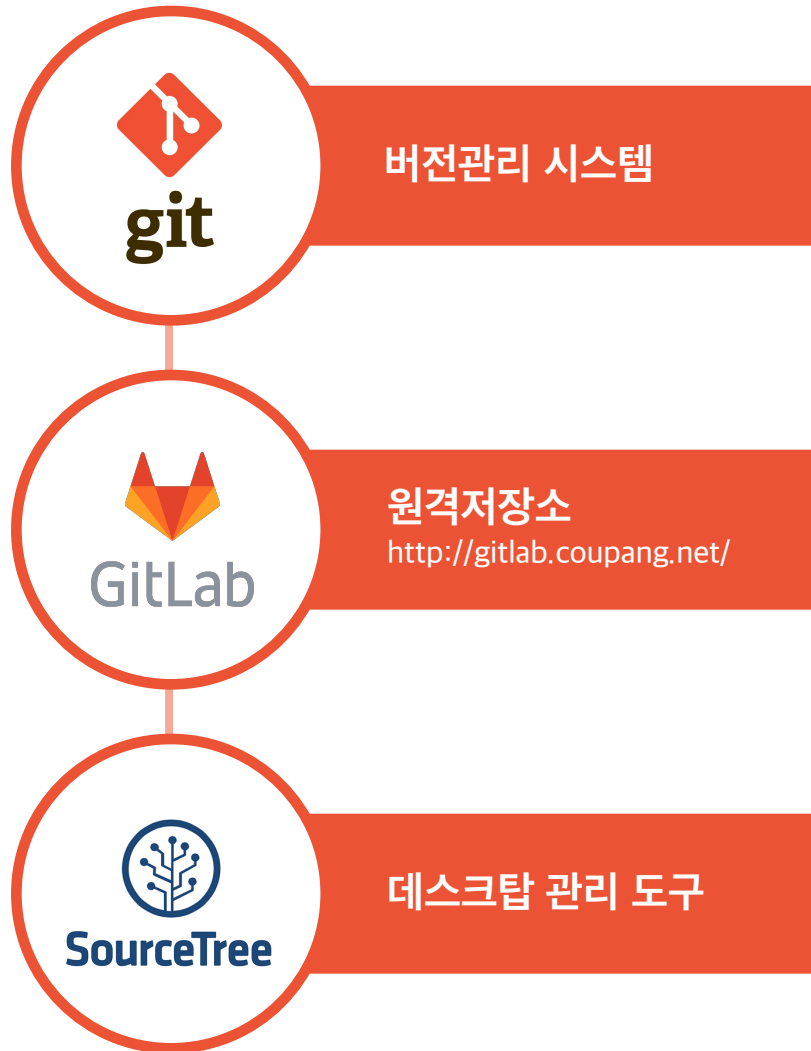


Search & Design



Designer Version Control System = **None** > Git

깃을 통해 모든 개발자와 디자이너의 소스가 버전이 관리된다.



깃랩 [: GitLab]

깃 프로젝트들을 위한 원격저장소 제공 서비스.
(오픈소스는 무료, 비공개 프로젝트는 유료)

특징

- 소스코드의 보안이 중요한 프로젝트에 적합
- 클라우드 / 서버 설치형 버전관리 지원
- 10명 이하 프로젝트 무료

유사 원격저장소 서비스



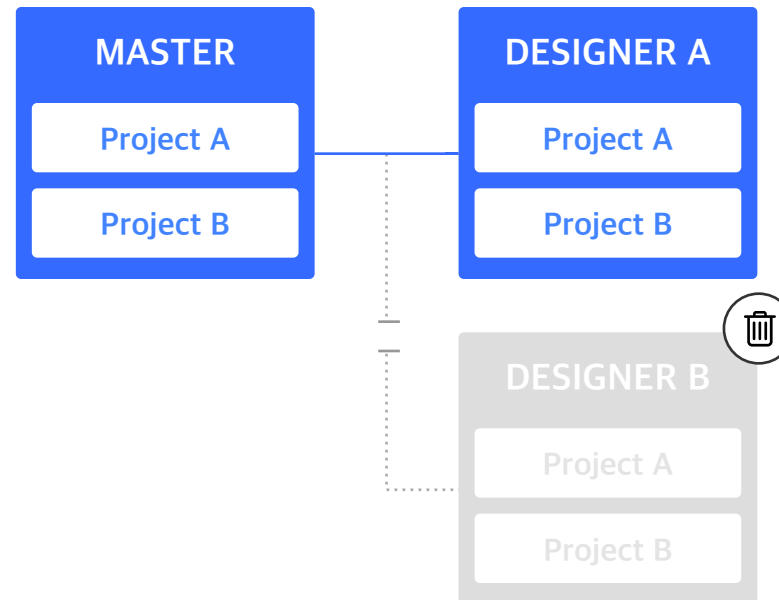
안정성

STABILITY

파일 유실 시, 버전관리 중인 파일 복구 가능.
최종 Release 버전 유지 최적화

ASIS

1. 소스의 유실 위험. (BackUp & History 관리 안됨)
2. UI 파편화가 발생하기 쉬운 구조



Copy & Paste UI

VCS

VERSION CONTROL SYSTEM

1. Fork : UI Source Copy & Paste.
2. Branch : Team/Designer/Project Separate.
3. Pull : Clone UI Source Sync.
4. Push : Update UI Source Upload.
5. CheckOut : Other UI Source Check
6. Merge Request : Merge Original UI Source

