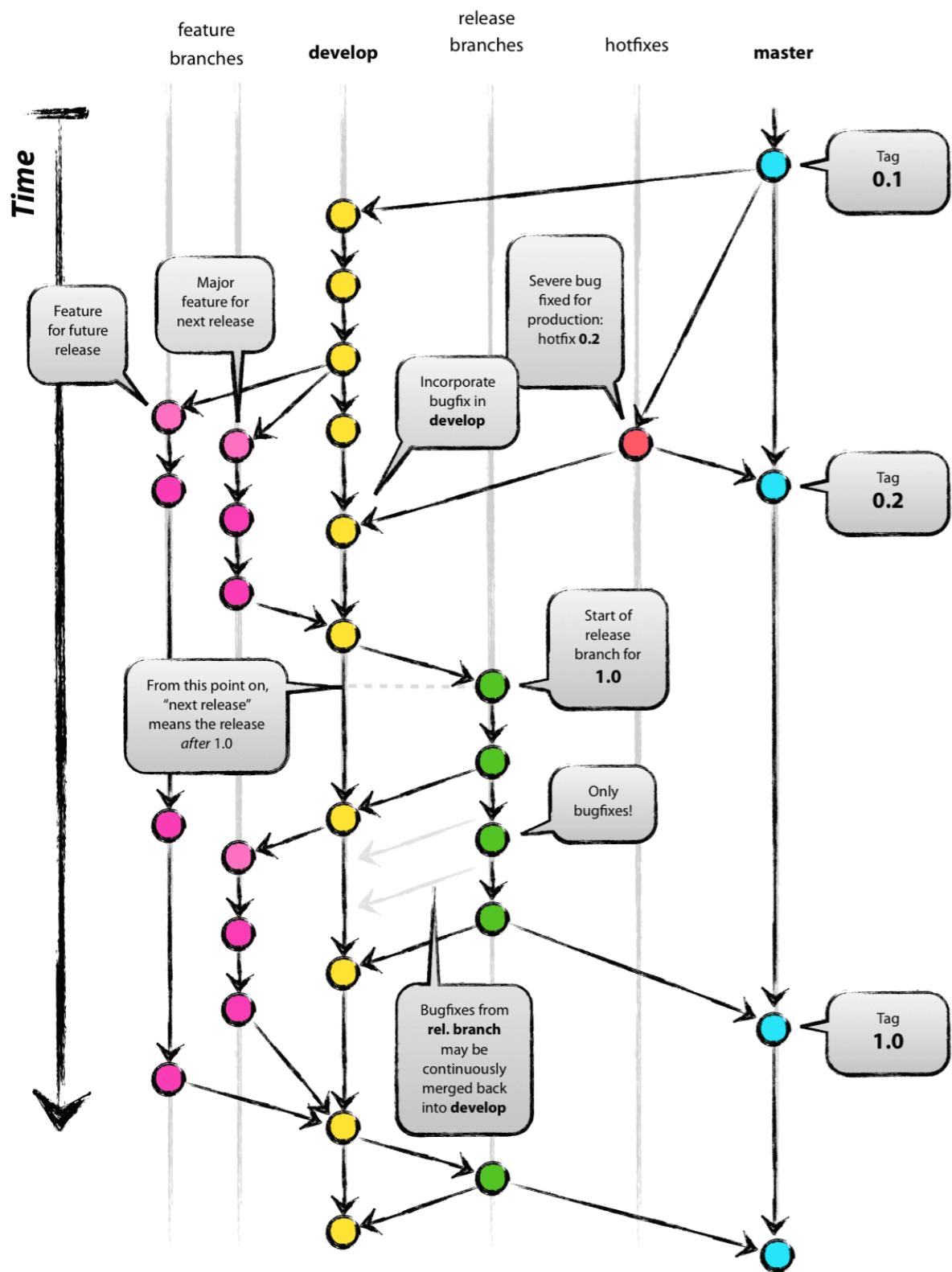




3강 - Git-Flow



1) 개념 정리

- **Git Flow**는 **Git**을 통해 효율적으로 협업할 수 있는 방법론 중 하나로, 현재 **Git**으로 개발할 때 거의 표준과 같이 사용되는 방법론이다.
- 서로간의 약속인 방법론이기에 각자 개발 환경에 따라 수정하고 변형해서 활용하면 된다.

Git Flow 브랜치 종류

- **master**: 기준이 되는 브랜치로 제품을 배포하는 브랜치
- **develop**: 개발 브랜치로 개발자들이 이 브랜치를 기준으로 각자 작업한 기능(**feature**)들을 합칩니다.(**merge**)
- **feature**: 단위 기능을 개발하는 브랜치로, 기능 개발이 완료되면 **develop** 브랜치에 합칩니다.
- **release**: 배포를 위해 **master** 브랜치로 보내기 전에 먼저 **QA**를 하기 위한 브랜치
- **hotfix**: **master** 브랜치로 배포를 했는데 버그가 생겼을 때 긴급 수정하는 브랜치

출처: UX 공작소

1. 일단 **master** 브랜치에서 시작을 합니다.
2. 동일한 브랜치를 **develop**에도 생성을 합니다. 개발자들은 이 **develop** 브랜치에서 개발을 진행합니다.
3. 개발을 진행하다가 회원가입, 장바구니 등의 기능 구현이 필요할 경우 **A**개발자는 **develop** 브랜치에서 **feature** 브랜치를 하나 생성해서 회원가입 기능을 구현하고 **B**개발자도 **develop** 브랜치에서 **feature** 브랜치를 하나 생성해서 장바구니 기능을 구현합니다.
4. 완료된 **feature** 브랜치는 검토를 거쳐 다시 **develop** 브랜치에 합칩니다.(**Merge**)
5. 이제 모든 기능이 완료되면 **develop** 브랜치를 **release** 브랜치로 만듭니다. 그리고 **QA(품질검사)**를 하면서 보완점을 보완하고 버그를 픽스합니다.
6. 모든 것이 완료되면 이제 **release** 브랜치를 **master** 브랜치와 **develop** 브랜치로 보냅니다. **master** 브랜치에서 버전추가를 위해 태그를 하나 생성하고 배포를 합니다.
7. 배포를 했는데 미처 발견하지 못한 버그가 있을 경우 **hotfixes** 브랜치를 만들어 긴급 수정 후 태그를 생성하고 바로 수정 배포를 합니다.