

# Software and systems engineering

Paulo Borba  
Informatics Center  
Federal University of Pernambuco

[phmb@cin.ufpe.br](mailto:phmb@cin.ufpe.br) ♦ [twitter.com/pauloborba](https://twitter.com/pauloborba)

# To do before class

- Watch video
- Read related part of chapter 10 in the textbook
- Send questions and opinions through slack

# Configuration management

Software and system  
development is often  
done in teams!

Team members  
contribute to the same  
artifacts

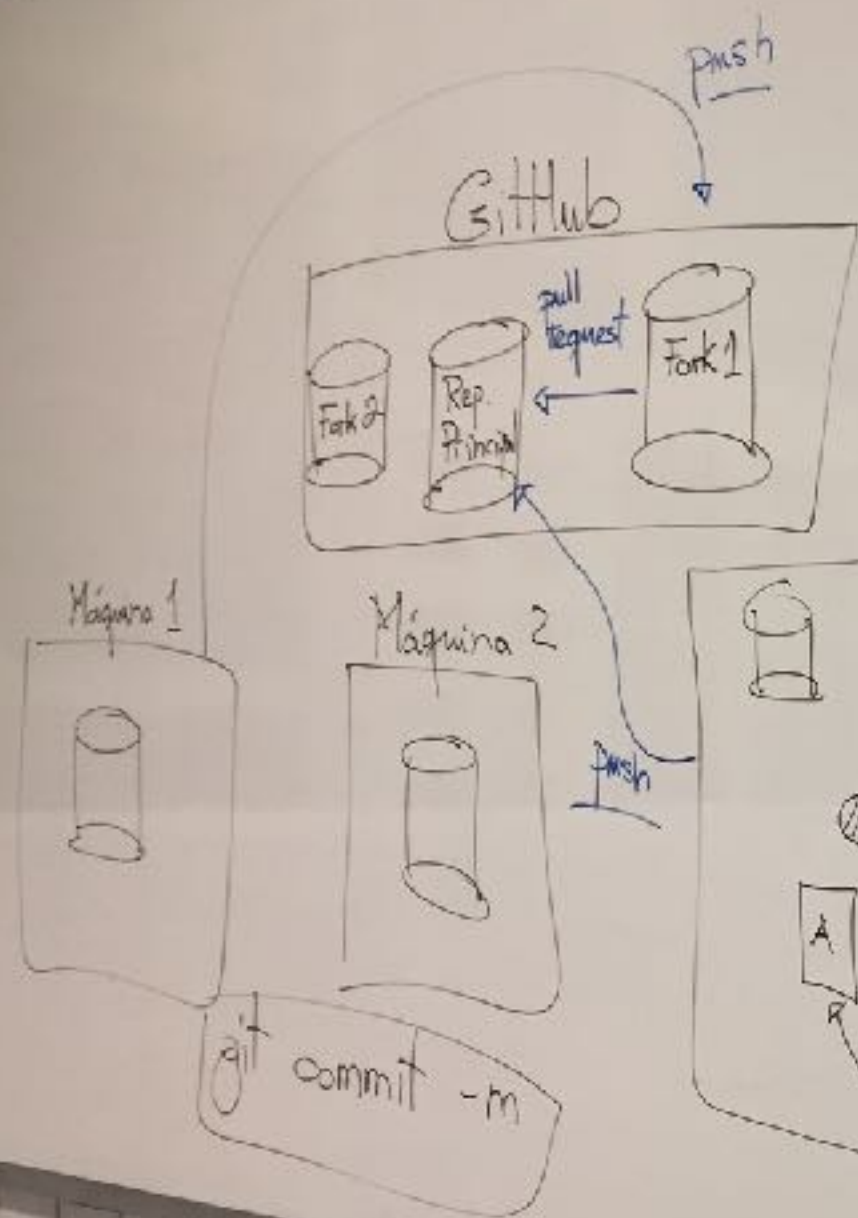
Productivity and quality  
problems sharing requirements,  
source code, etc. via...

- email
- dropbox
- google docs

Source code (artifact)  
control systems

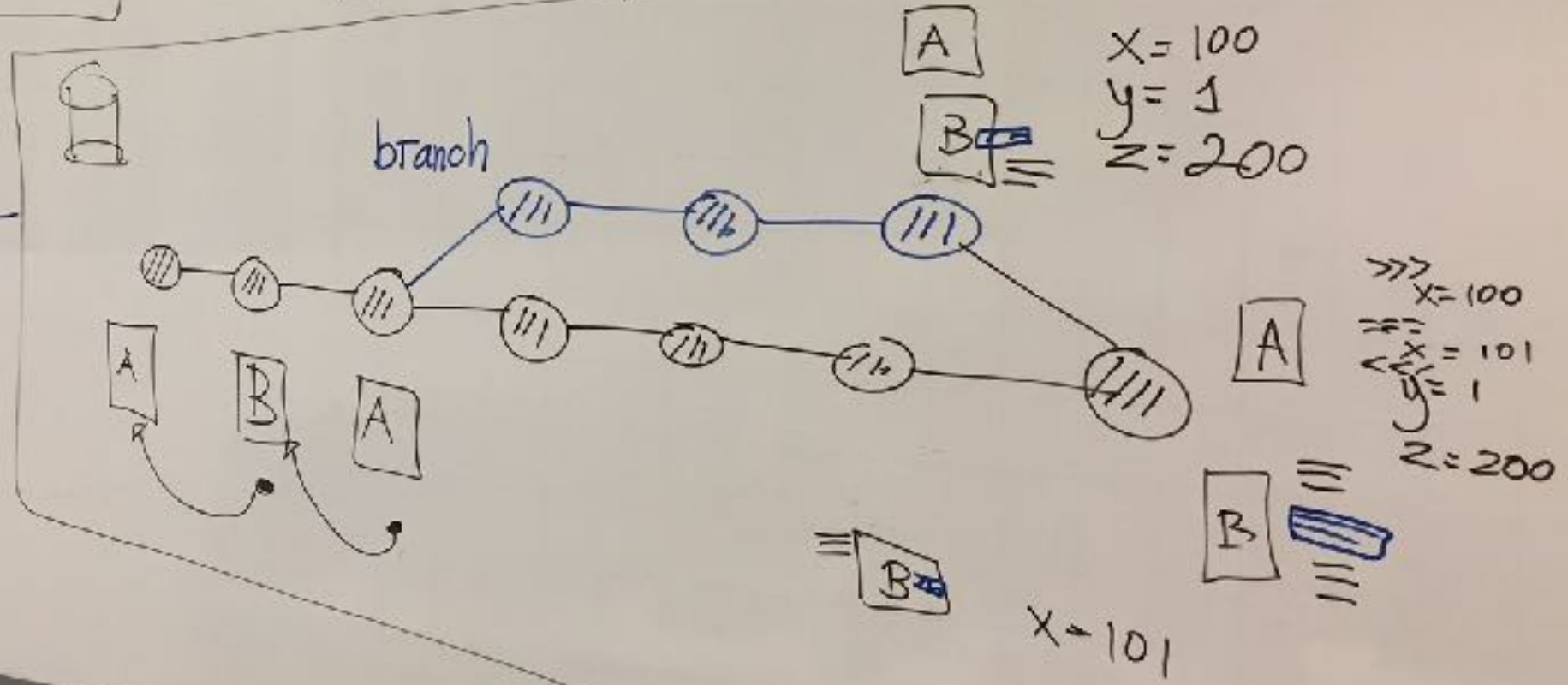
# Overview of main concepts



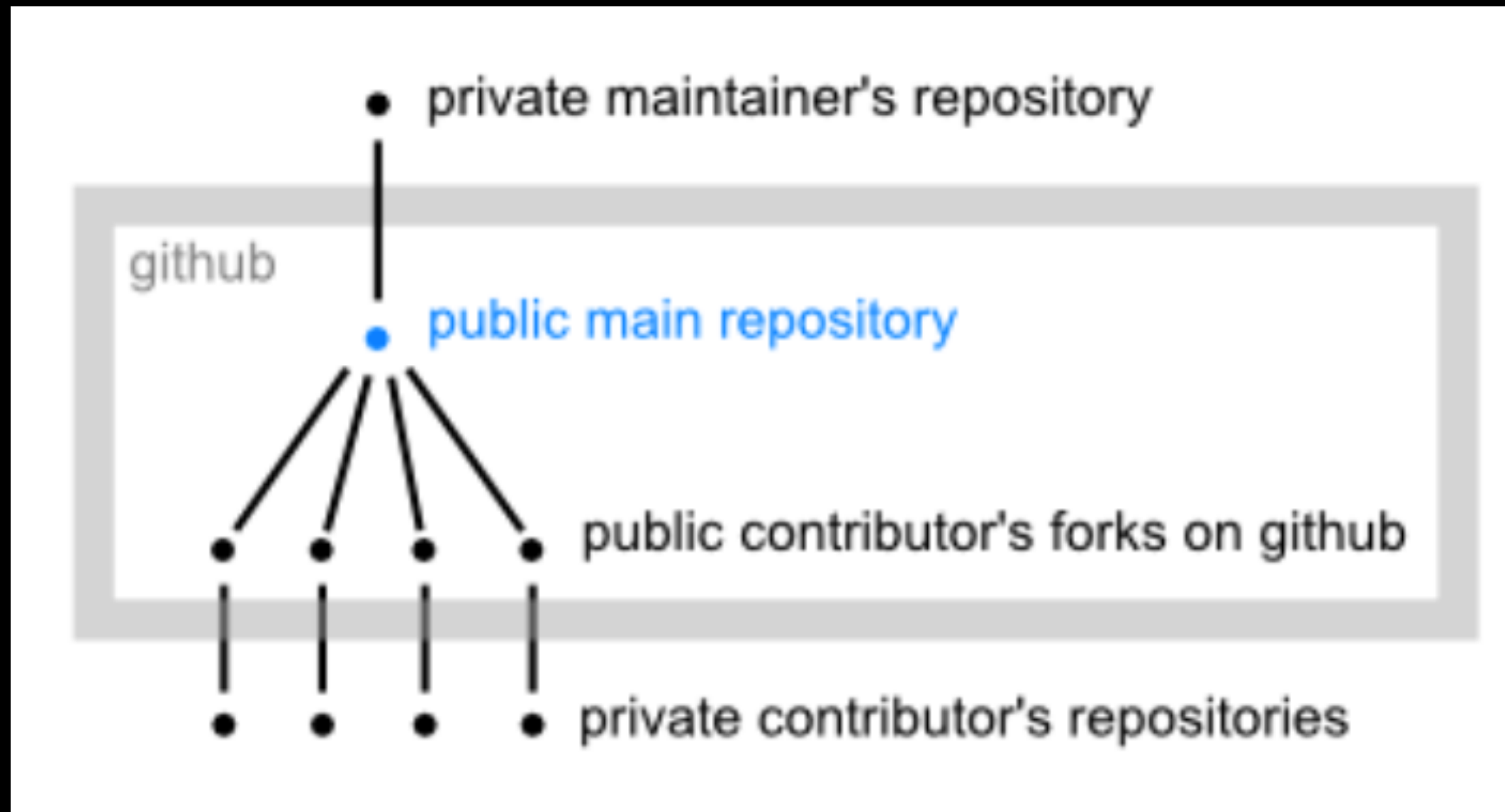


Minha máquina

merge

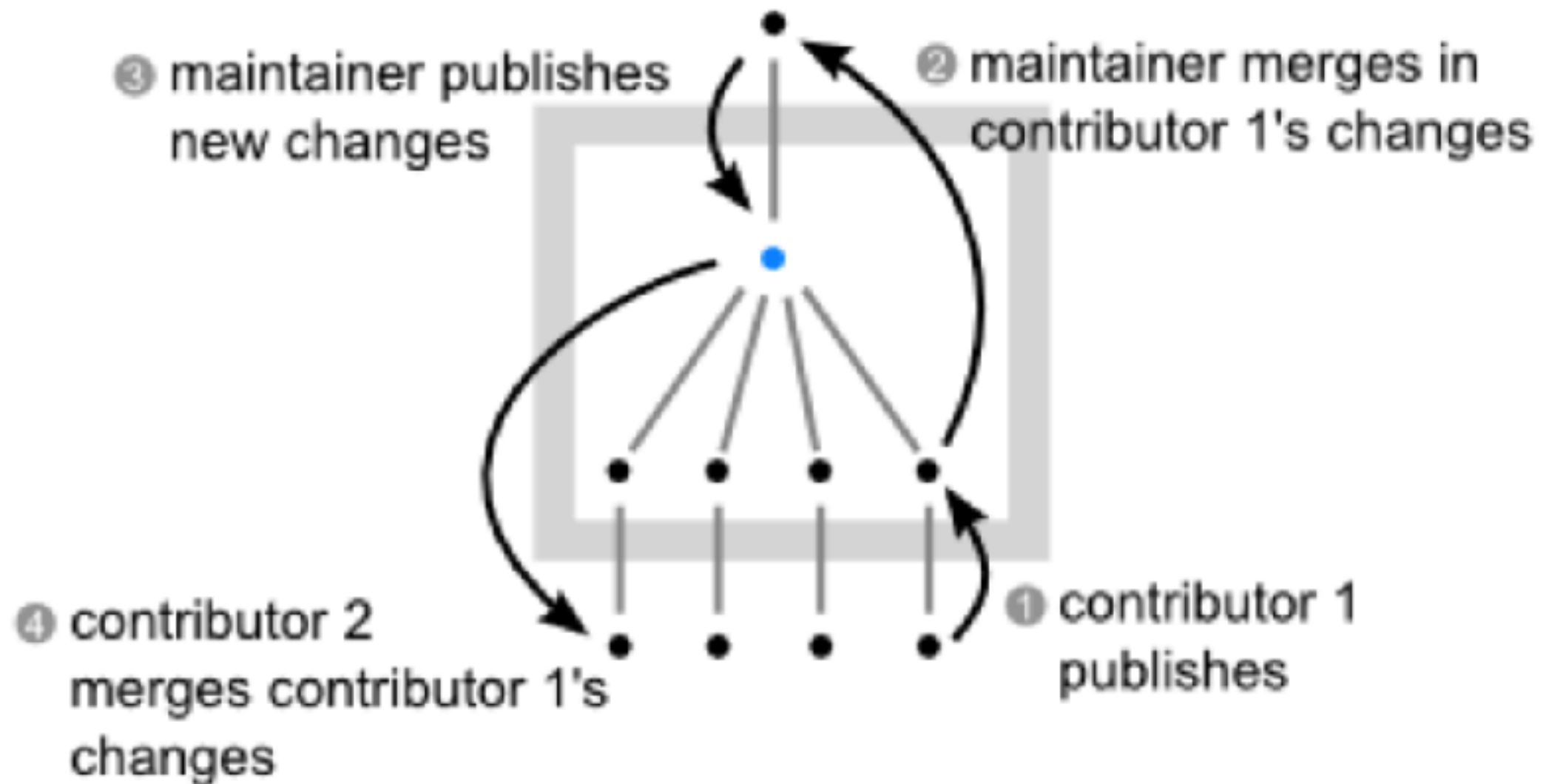


# Forking and cloning



From [http://www.eqqon.com/index.php/Collaborative\\_Github\\_Workflow](http://www.eqqon.com/index.php/Collaborative_Github_Workflow)

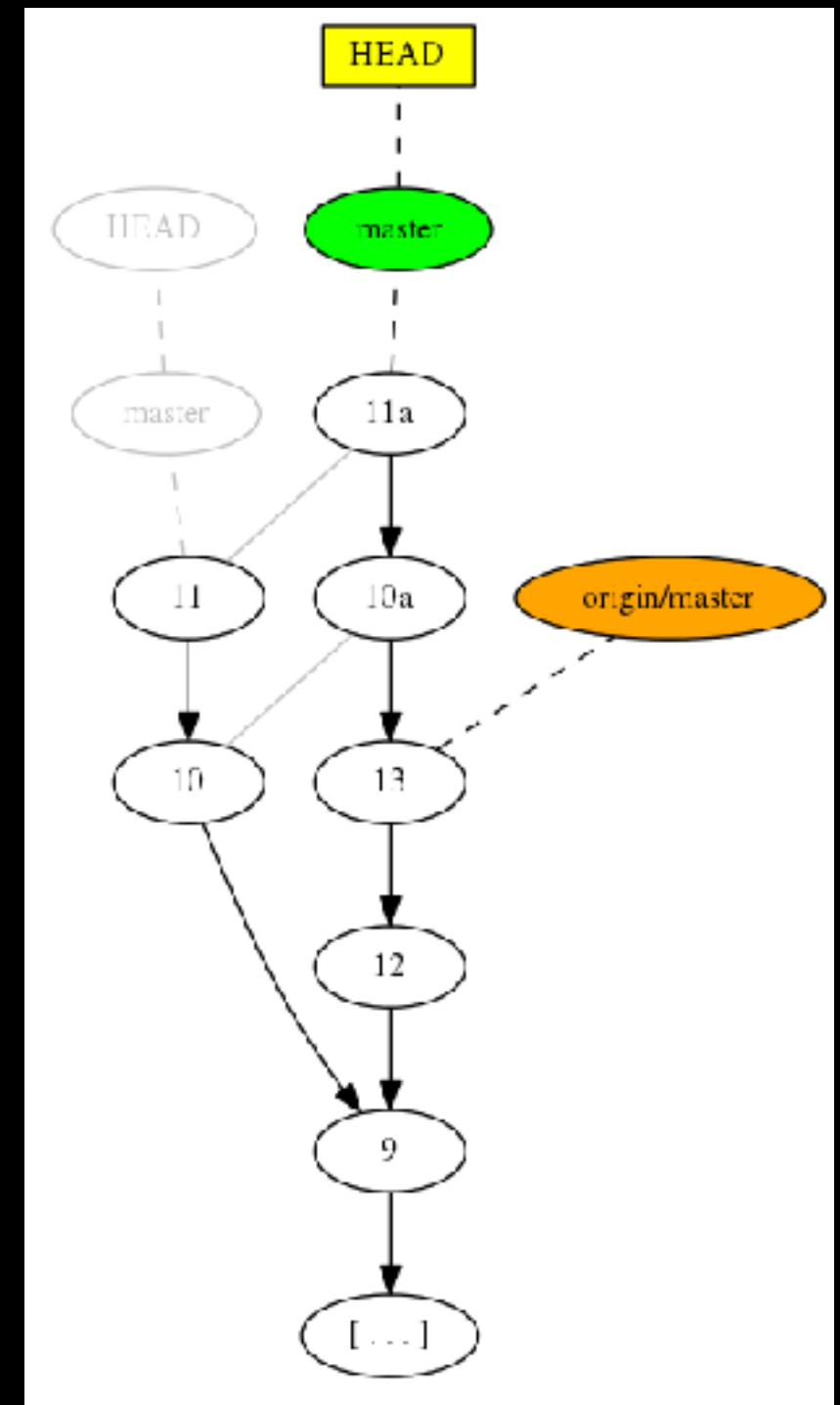
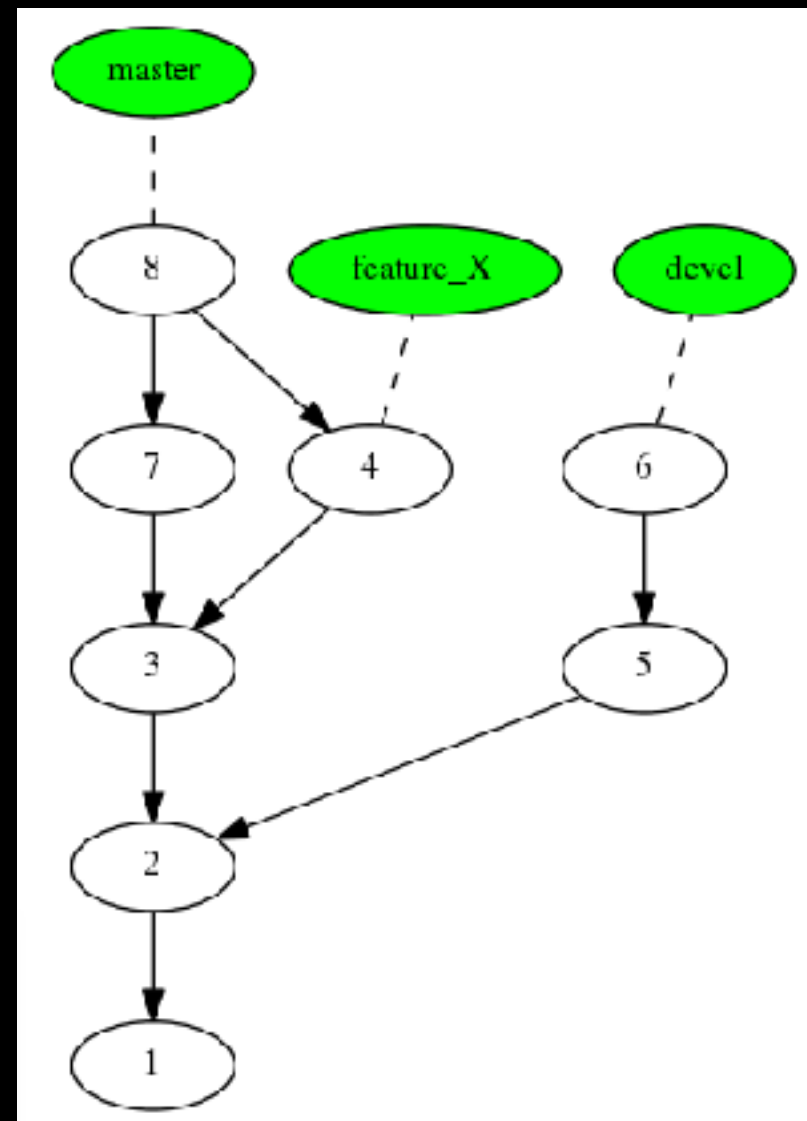
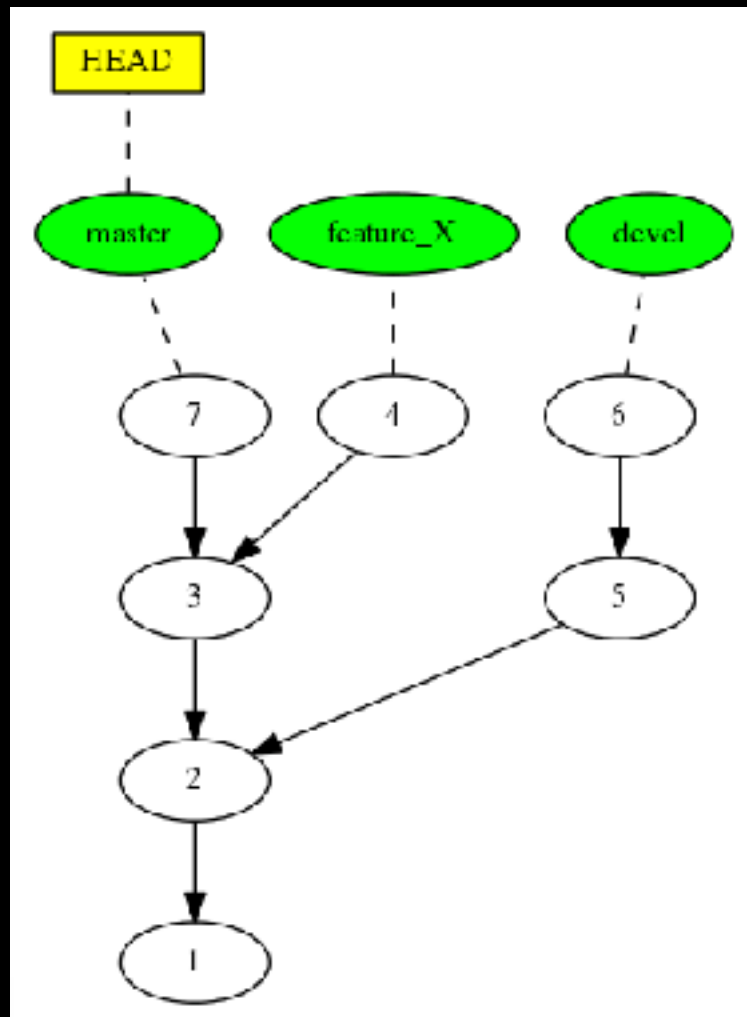
# Pushing, requesting pulling, and merging



Take notes,  
now!

Hands-on!

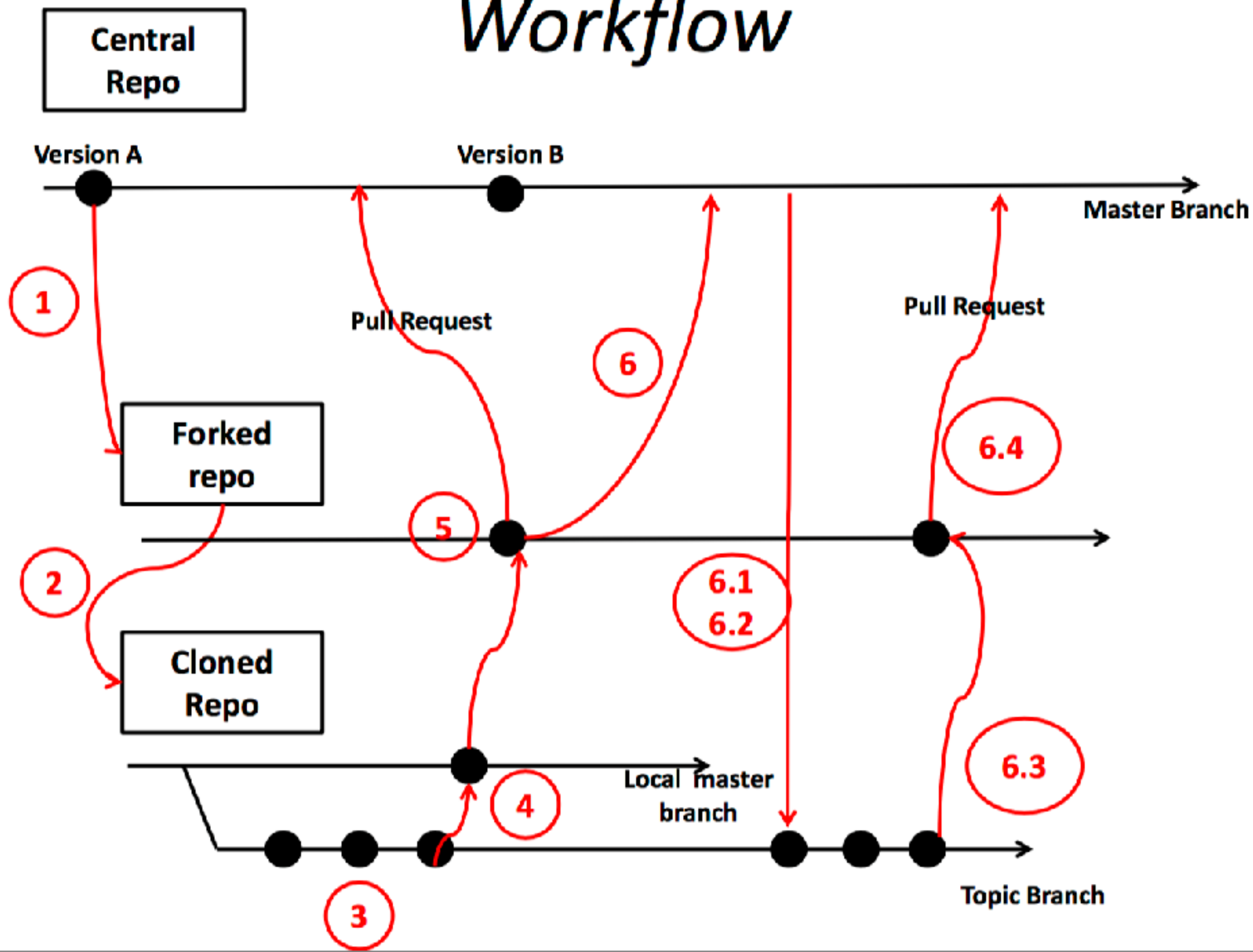
# Branches, merging, rebasing and squashing



# Reasons for creating branches

- Feature or component development
- Bug fixing in previous releases
- Customer specific variants

# Workflow





# Before integrating...

- Review scenarios and features, or write scenarios in pairs
- Make sure they follow style and quality criteria
- Organize review procedures and meetings

# Checklist

# Checklist

- Clear commit messages, exactly corresponding to what has changed
  - no language errors
- Commit early and often
  - avoid tangled commits

# Checklist

- Antes de enviar um pull request, deve-se, obrigatoriamente, fazer um merge ou rebase para que suas modificações sejam aplicadas à versão mais atual do repositório central do seu projeto, não à última versão que você tinha na sua máquina e com informações desnecessárias.
- A mensagem do pull request deve descrever com precisão e clareza o seu conteúdo.
- Cada pull request deve incluir somente os arquivos que foram alterados para a atividade.

# Checklist

- Pull requests não devem conter arquivos gerados ou de configuração particular de cada máquina ou IDE, como `.classpath`, `.project`, `.class`. O mesmo vale para arquivos de log e similares. Não deve-se dar commits nesses arquivos. Use o `.gitignore` para evitar esse problema.
- Evite pular linhas, adicionar espaços, etc. em partes do código que você não precisa modificar; cada mudança desses precisa ser analisada por quem vai integrar a sua contribuição ao repositório principal, e podem gerar conflitos.
- Tenha certeza de que todos os testes estejam passando na versão incluída no pull request.

Continuous integration,  
continuous deployment

# Change requests and change managements

# Issues or change requests

- Bug report
- Need for new feature or scenario
- Need for changing a particular feature
- Need for code improvement
- Need for performance improvement
- Chore description (provide no direct value to the customer)



Take notes,  
now!

# CM research at CIn

- Advanced merging tools: Paulo
- Collaboration conflicts characteristics: Paulo

# To do after class

- Answer questionnaire (check classroom assignment), study correct answers
- Finish exercise (check classroom assignment), study correct answers
- Read, again, part of chapter 10 in the textbook
- Evaluate classes
- Study questions from previous exams

# To do after class

- Use Pro Git, and Git Cheat Sheet as references
- Read either Git concepts simplified or A Visual Git Reference
- Opcionalmente, ler artigo que descreve um workflow colaborativo, e outro artigo que adiciona mais alguns detalhes
- Opcionalmente, instale e use uma ferramenta que ajuda na resolução de conflitos

# Questions from previous exams

- Explique brevemente (a) o que é um branch em um sistema de controle de versão, e (b) porque você criaria um em seu projeto. Explique também (c) o propósito e o funcionamento da operação de merge.
- Podemos tanto fazer merge de mudanças do branch principal em um branch de feature (atualizando o branch de feature) quanto fazer merge de mudanças de um branch de feature no branch principal (atualizando o branch principal)? Explique brevemente sua resposta.

# Software and systems engineering

Paulo Borba  
Informatics Center  
Federal University of Pernambuco

[phmb@cin.ufpe.br](mailto:phmb@cin.ufpe.br) ♦ [twitter.com/pauloborba](https://twitter.com/pauloborba)