The tale of a model **hoarder**

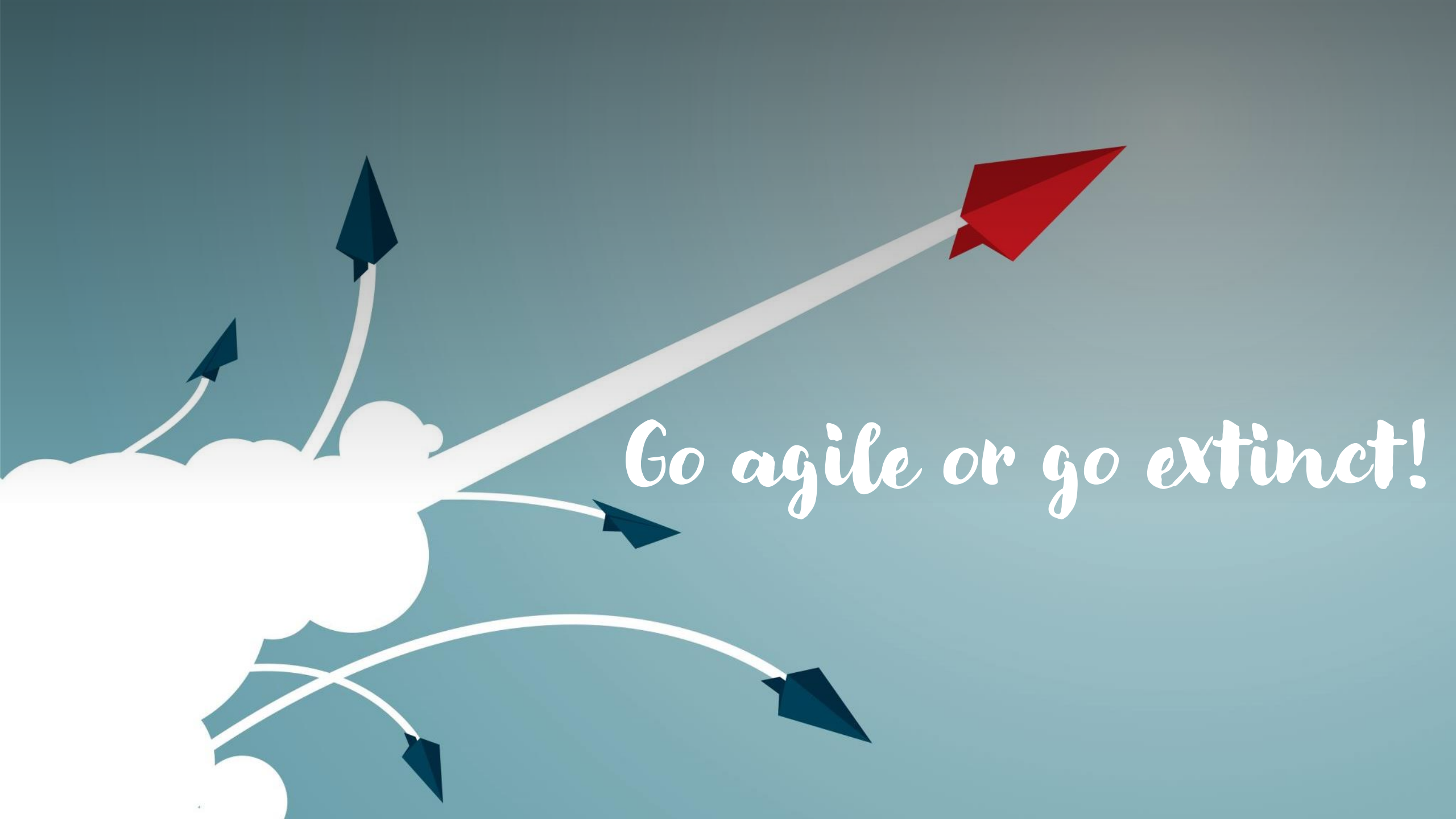
Agile Archimate? Using Enterprise Architect and GitHub to craft your models!

LieberLieber Software
Daniel Siegl
11.09.2024

Daniel Siegl

- Involved with Model Based Engineering Tools since 1999
- Business “Development” at Syntevo and LieberLieber
- INCOSE Deputy Director for Standards
- Chaired PROSTEP Workgroups
- Involved with OMG since 2008

#MBSE
#Collaboration
#Interchange
#Standards
#git #github

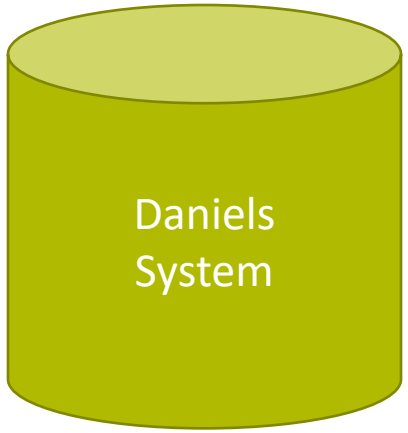

Go agile or go extinct!

LemonTree 4.0

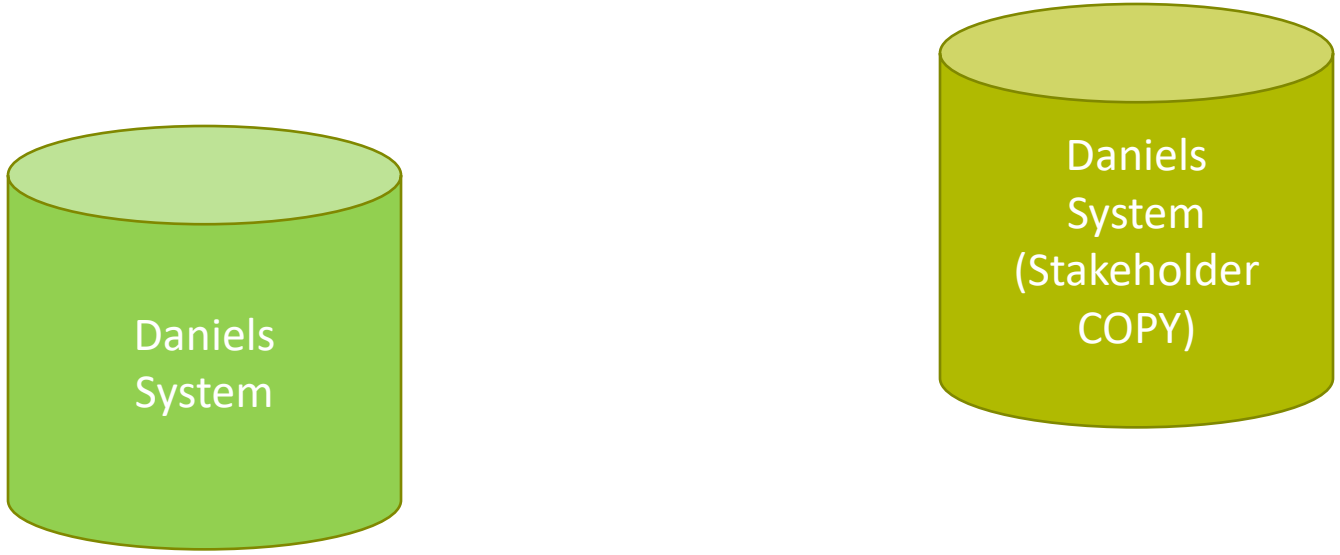
LemonTree©
Fresh Model Versioning

Model Based Team collaboration using strategies from Software Engineering.

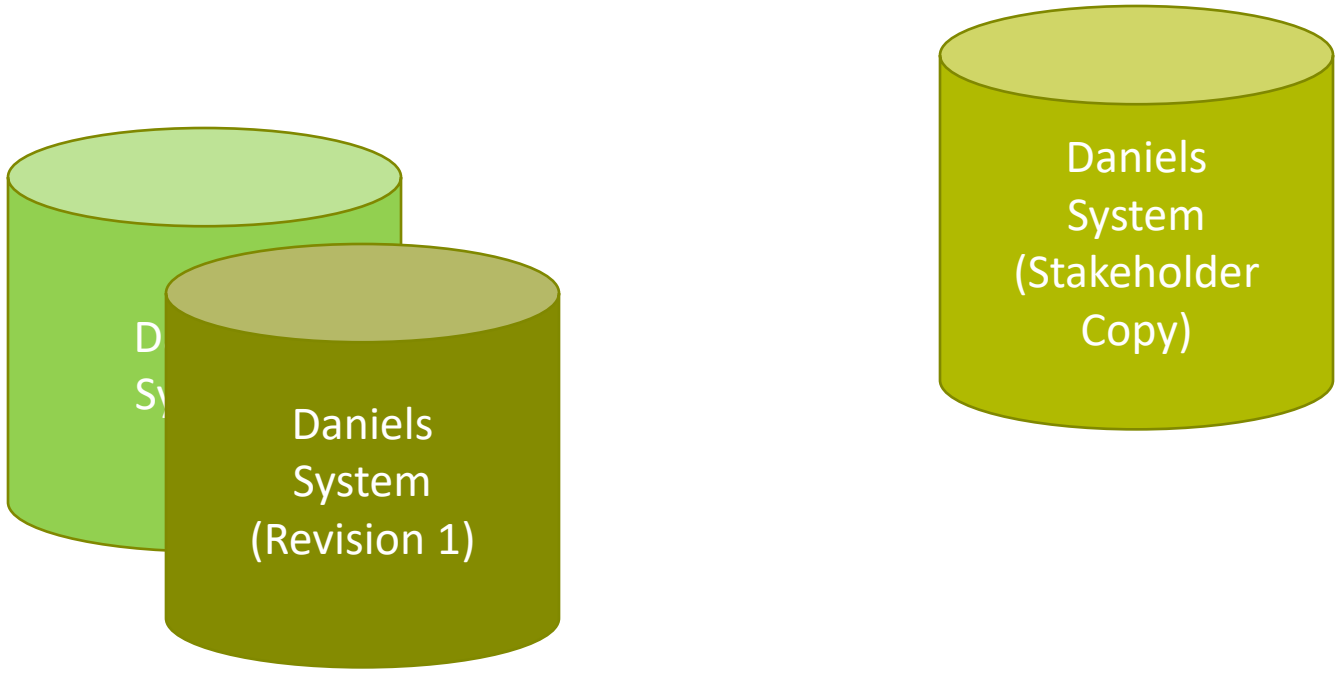
I create a Model



I am done – testers wants a copy
I start an update

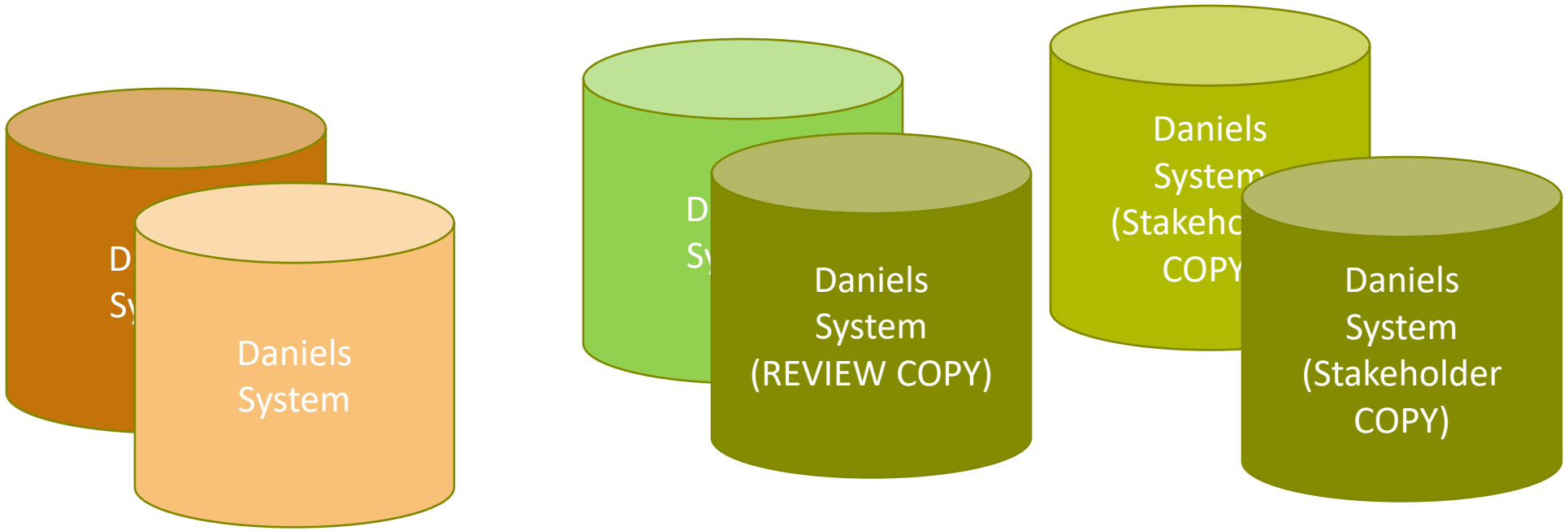


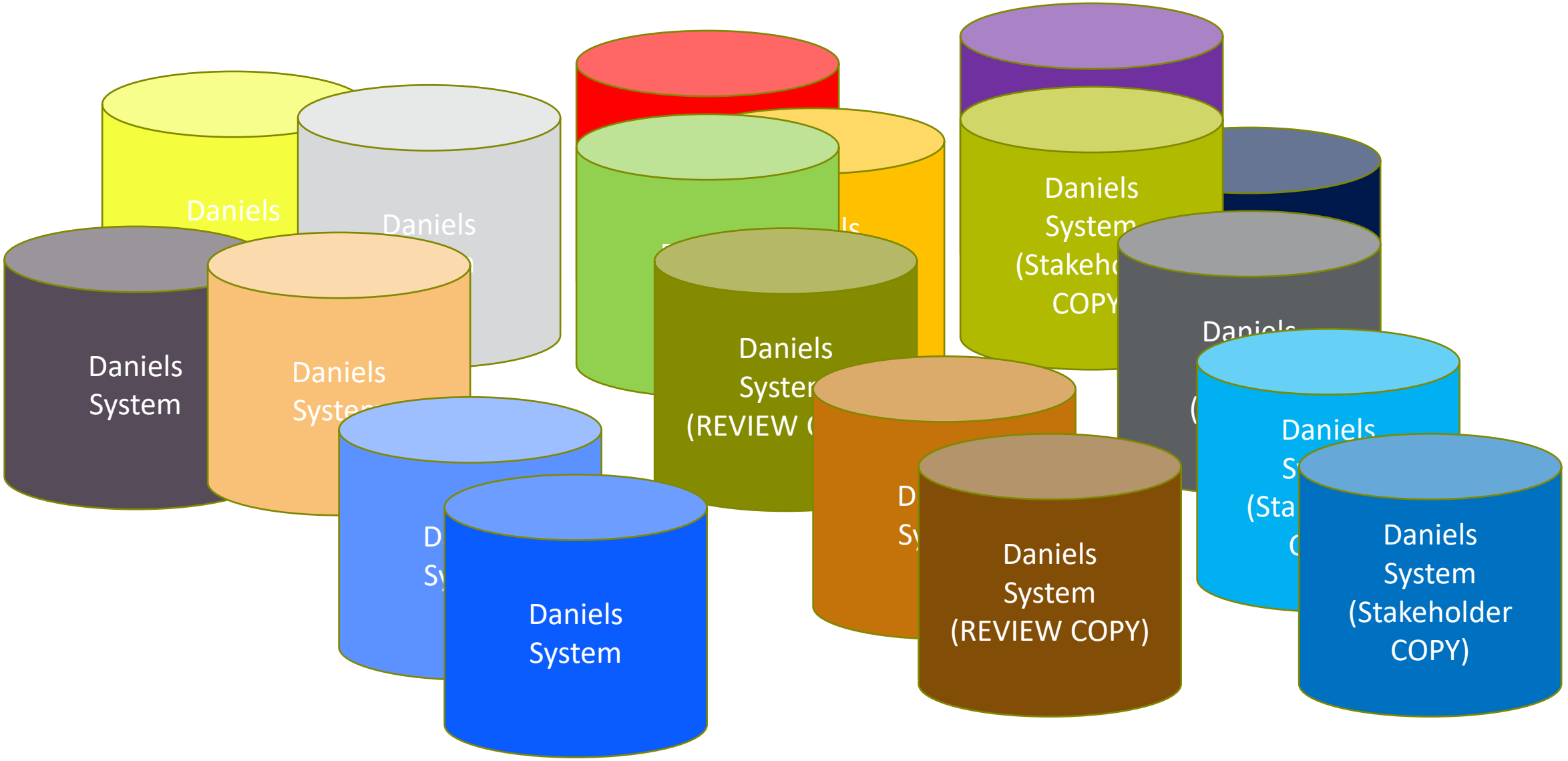
I need to do a variation for some one else



They got an IT too

And now they want to review the stuff I did




Daniels
Daniels
Daniels
System
Daniels
Syste
Daniels
Syste
(REVIEW
Daniels
System
(Stakeho
COPY
Daniels
S
(Sta
C
Daniels
System
(REVIEW COPY)
Daniels
System
(Stakeholder
COPY)

One truth? Really? – History?

Problem Statement

- „Trunk“ only not feasible for systems
- Agility also needed in MBSE to meet goals
- **Complexity** of Cyber-Physical Systems
- **Manage Model quality in distributed Teams**

- **Cross company projects on shared IP**
- **Digital Twin – and the dead meat…**

Modeling Collaboration User Requirements

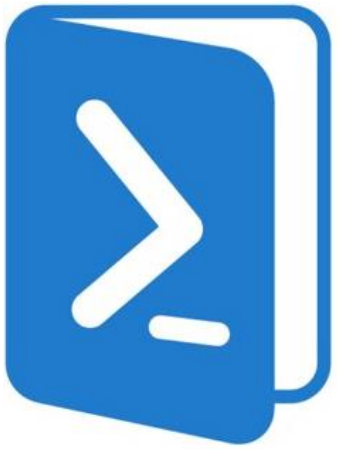
- **It should be easy to release Models**
- It shall be possible **to view and modify past versions** of the Model
- It shall be possible to evaluate alternative scenarios
- **It shall be possible to give new users a sandbox**
- Stake holders shall only be able to access „Staging/Review“ and „Production“ Models
- Stake Holders need to be able to consume and review the model via a Web browser

I need a ROBOT

Engineers have Robots!

„**The first real version control system goes back to 1972** when a Source Code Control System (SCCS) started getting developed by Marc J. Rochkind at Bell laboratories as a set of commands developed for OS/MVT, and later on UNIX.”

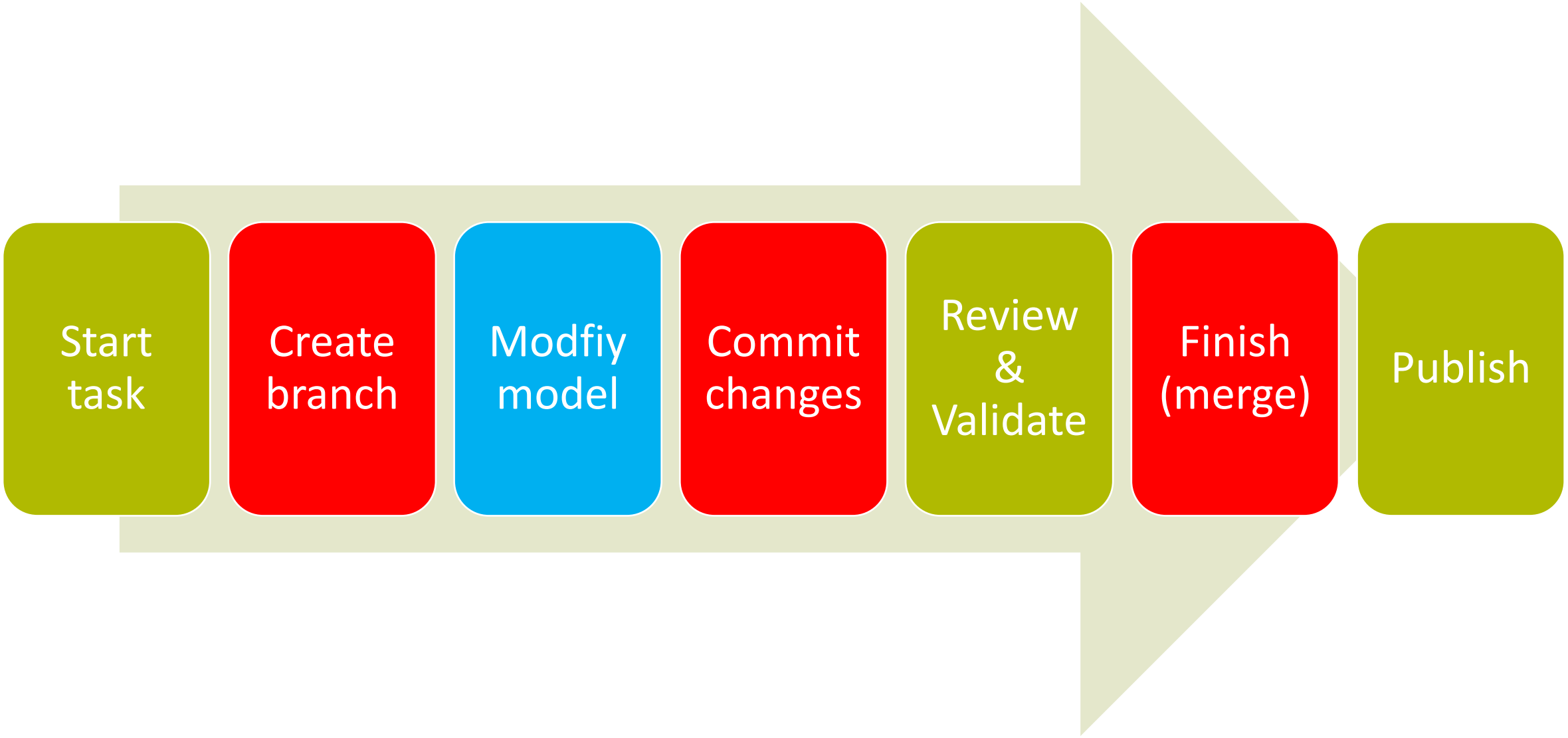
Jenkins

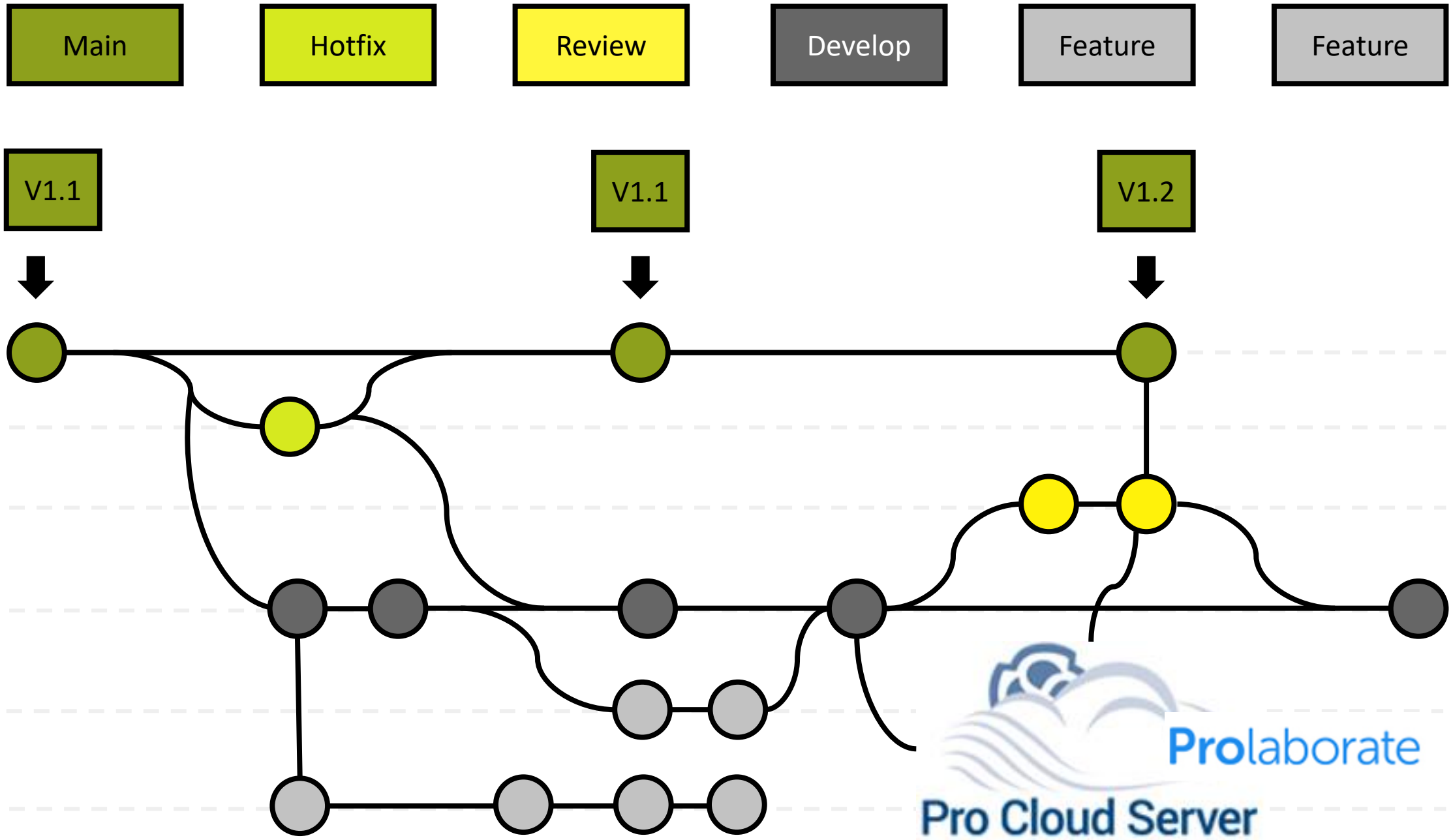


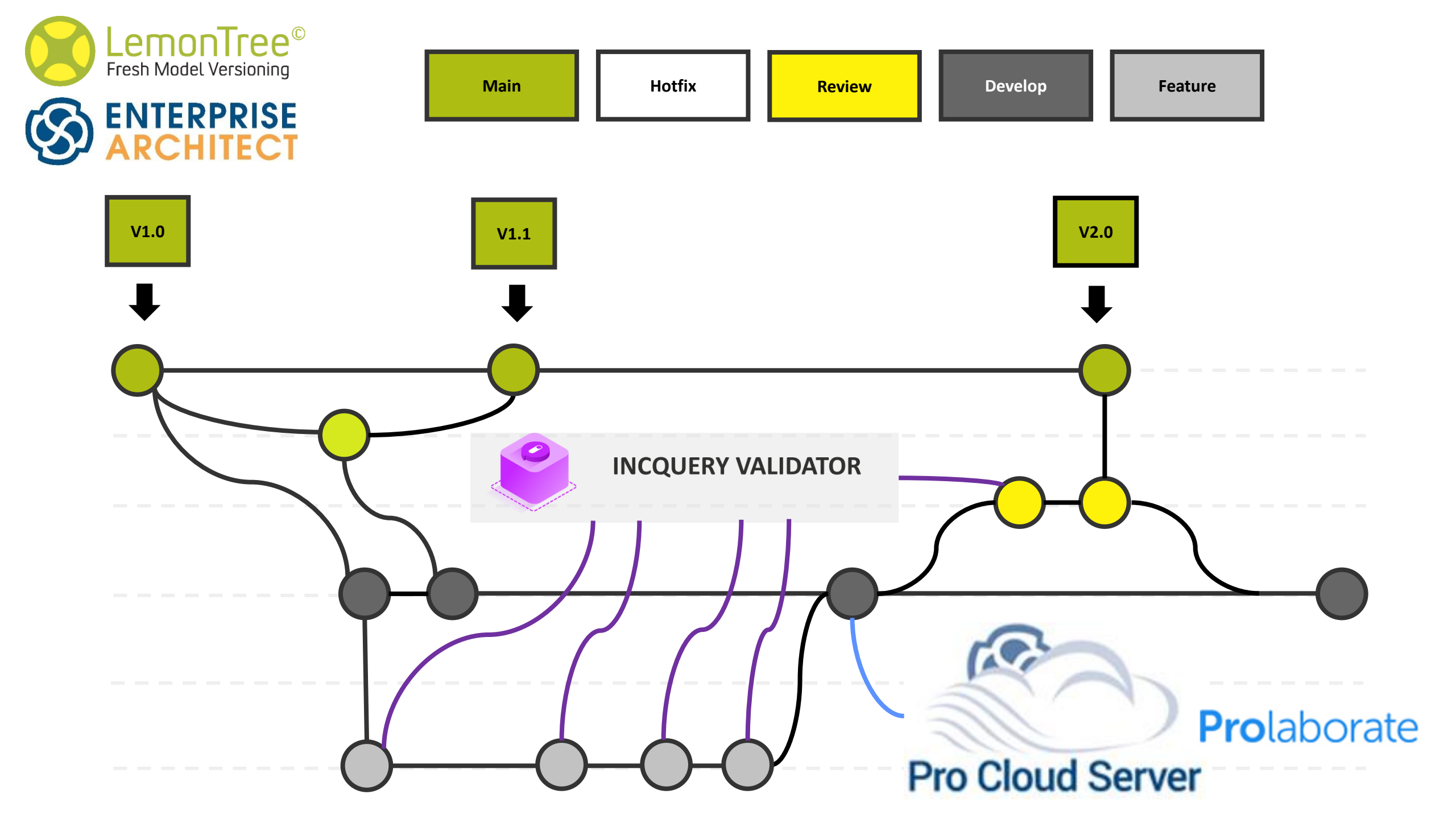
PowerShell

GitHub

Some of our 200+ global Users….

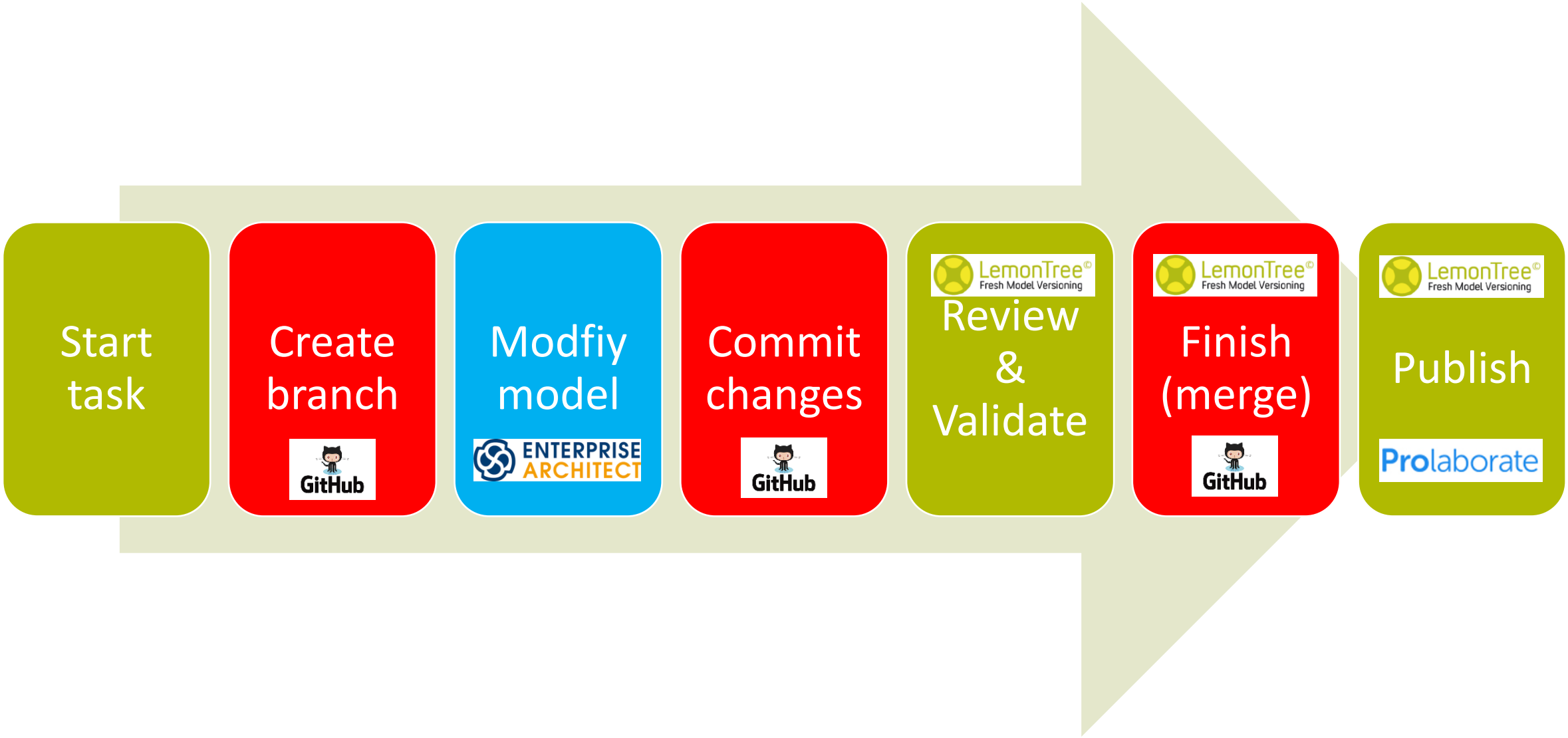

Start task
Create branch
Modfiy model
Commit changes
Review & Validate
Finish (merge)
Publish


Main
Hotfix
Review
Develop
Feature
Feature
V1.1
V1.1
V1.2
Prolaborate
Pro Cloud Server


LemonTree©
Fresh Model Versioning
ENTERPRISE ARCHITECT
Main
Hotfix
Review
Develop
Feature
V1.0
V1.1
V2.0
INCQUERY VALIDATOR
Prolaborate
Pro Cloud Server

DON‘T SAY AGILE

IF YOU CAN‘T MERGE YOUR MODELS


Start task
Create branch
GitHub
Modfiy model
ENTERPRISE ARCHITECT
Commit changes
GitHub
LemonTree© Fresh Model Versioning
Review & Validate
LemonTree© Fresh Model Versioning
Finish (merge)
GitHub
LemonTree© Fresh Model Versioning
Publish
Prolaborate

Client

Local Git Repository

.qea (x) file

Git Binaries
+ Client

LemonTree

Push

Pull

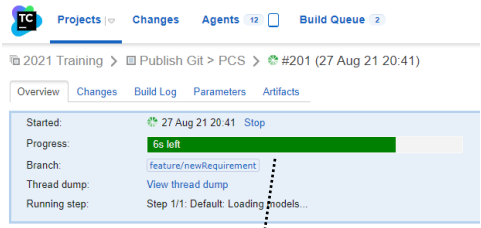
Server

Git Server
(f.e. GitHub)

.qeax file

Trigger

Build Server
TeamCity/Jenkins/GitHub

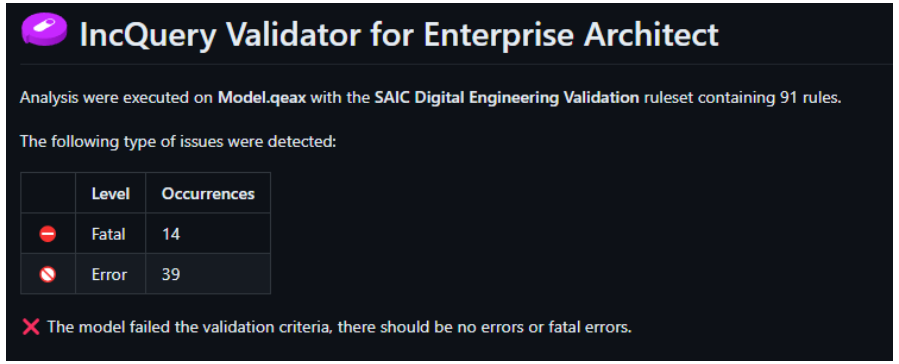


LemonTree Automation

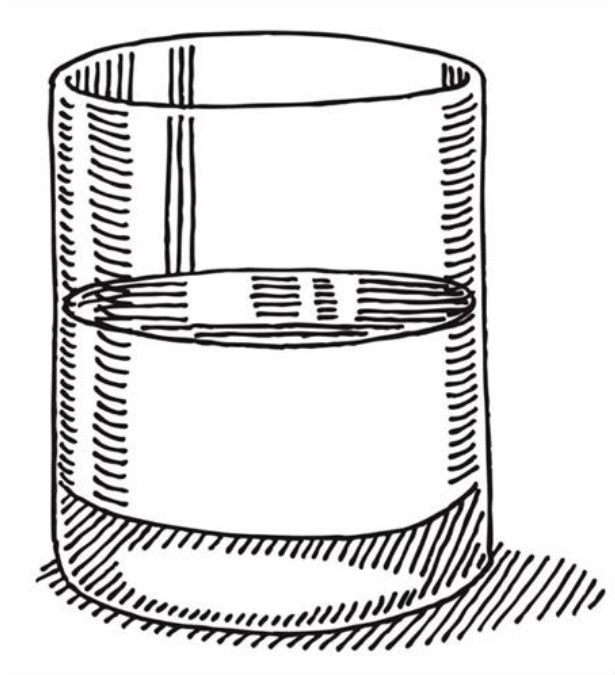
Prolaborate

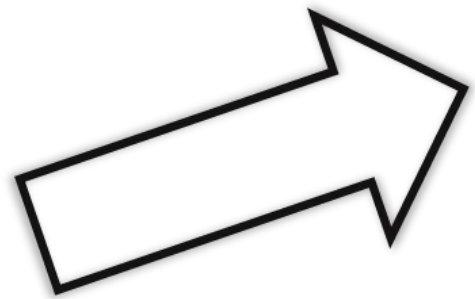
Web Access

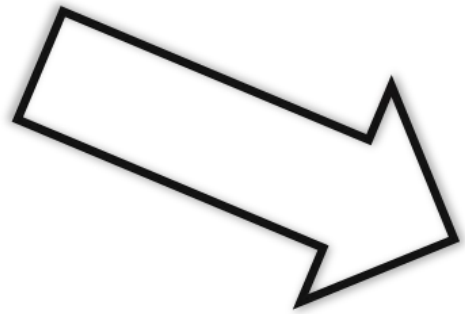
Pro Cloud Server

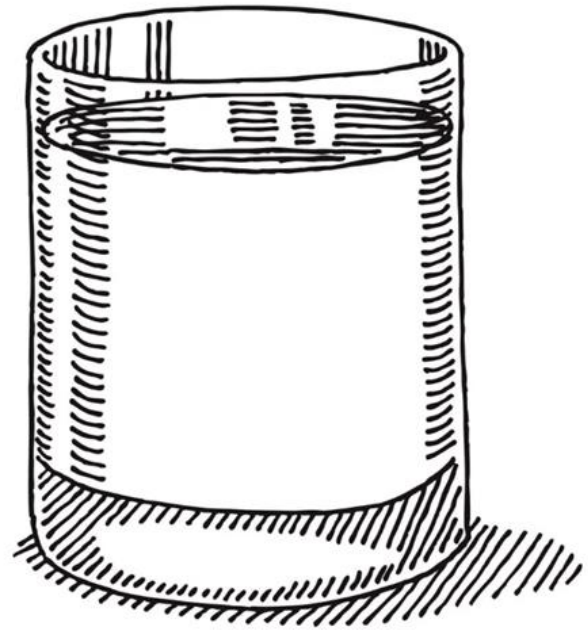


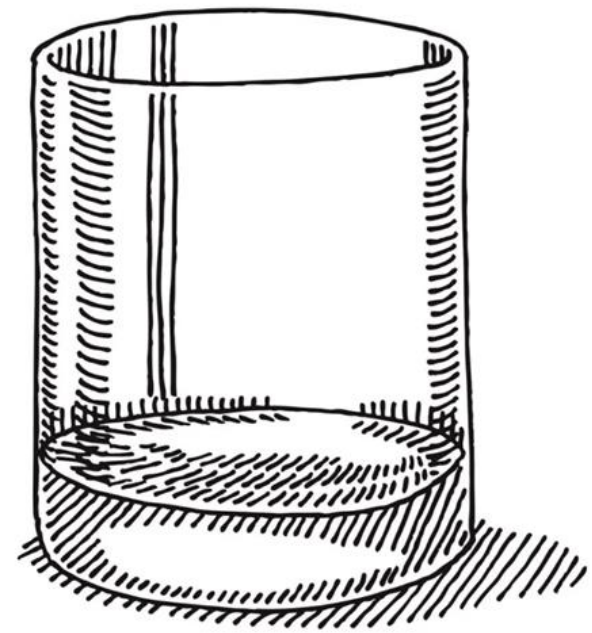
DEVOPS ISN'T IT?











„Social“ Benefits

- New users – can work in a sandbox – **easy to mentor and review**
- Only **„finished“ models are published** for Stakeholders/Reviewers
- Easy to **revert changes**
- If a Feature is discarded/postponend there will be **no „dead meat“ in the „main“ model**
- Culture of visible measuring **improves quality and transparency**

Conclusion

- Established workflows from SE for modeling
- **Agile team collaboration based on branching**
- **Agile (cross) team collaboration based on Model Components**
- Consistent development state due to precise versioning
- Perfect triggers for peer reviews
- Publish/Release consistent models to Stakeholders

#XMIISDEAD

LemonTree 4.0

LemonTree©
Fresh Model Versioning

Model Based Team collaboration using strategies from Software Engineering.