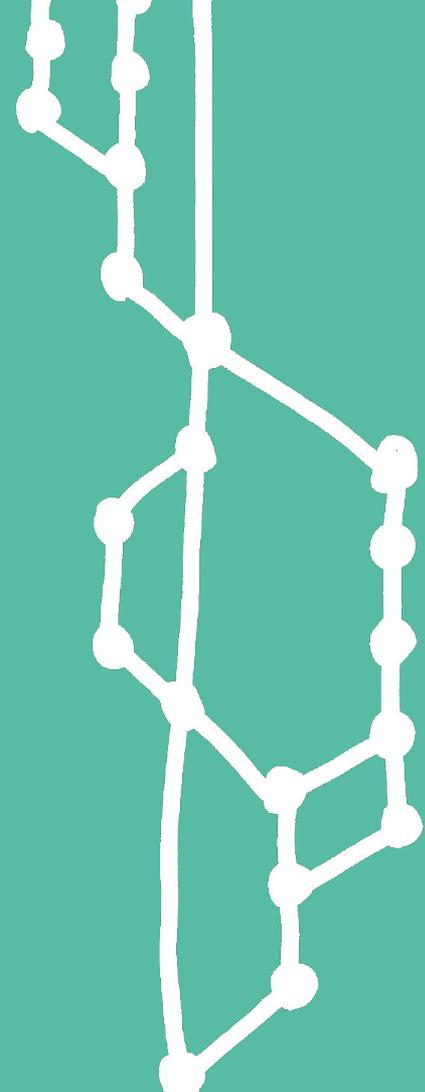
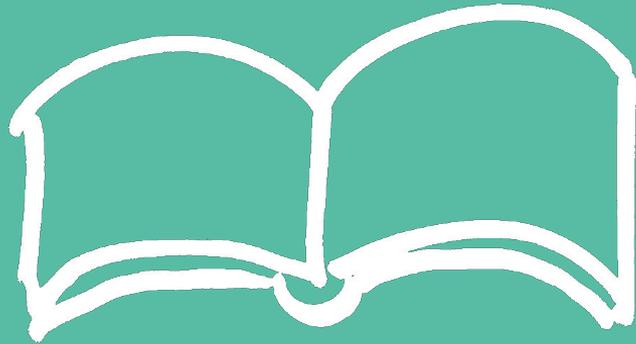


On Death and Dying:



Moving to
Trunk Based Development

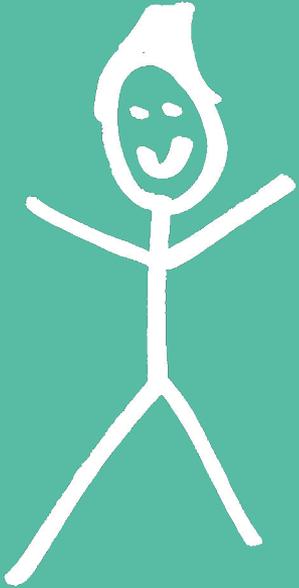




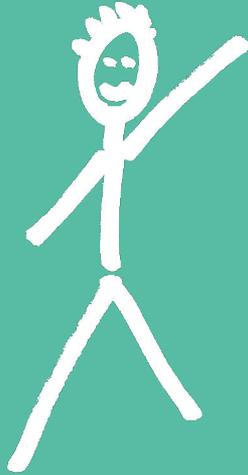
Dr. Elizabeth Kübler-Ross,
On Death and Dying, 1969



Dionysus Team



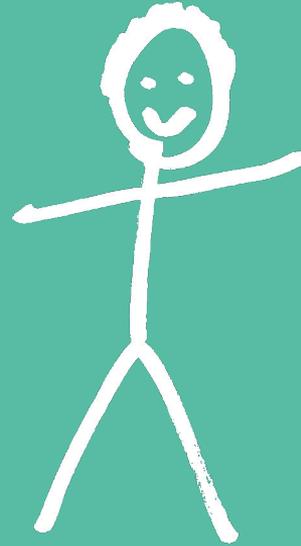
Siphon



Julio



Nora

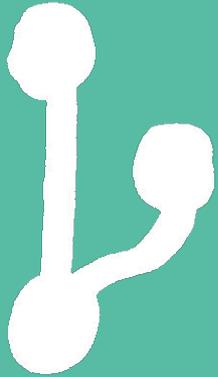


Andreas

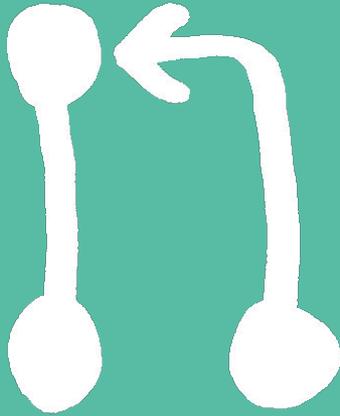


Fei-Fei

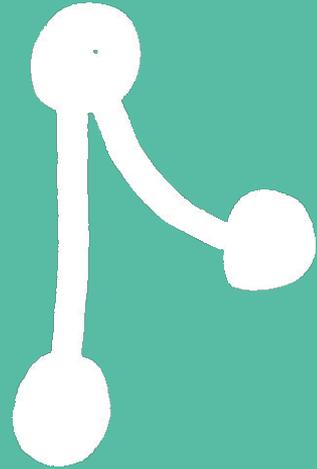
Github Flow



Fork



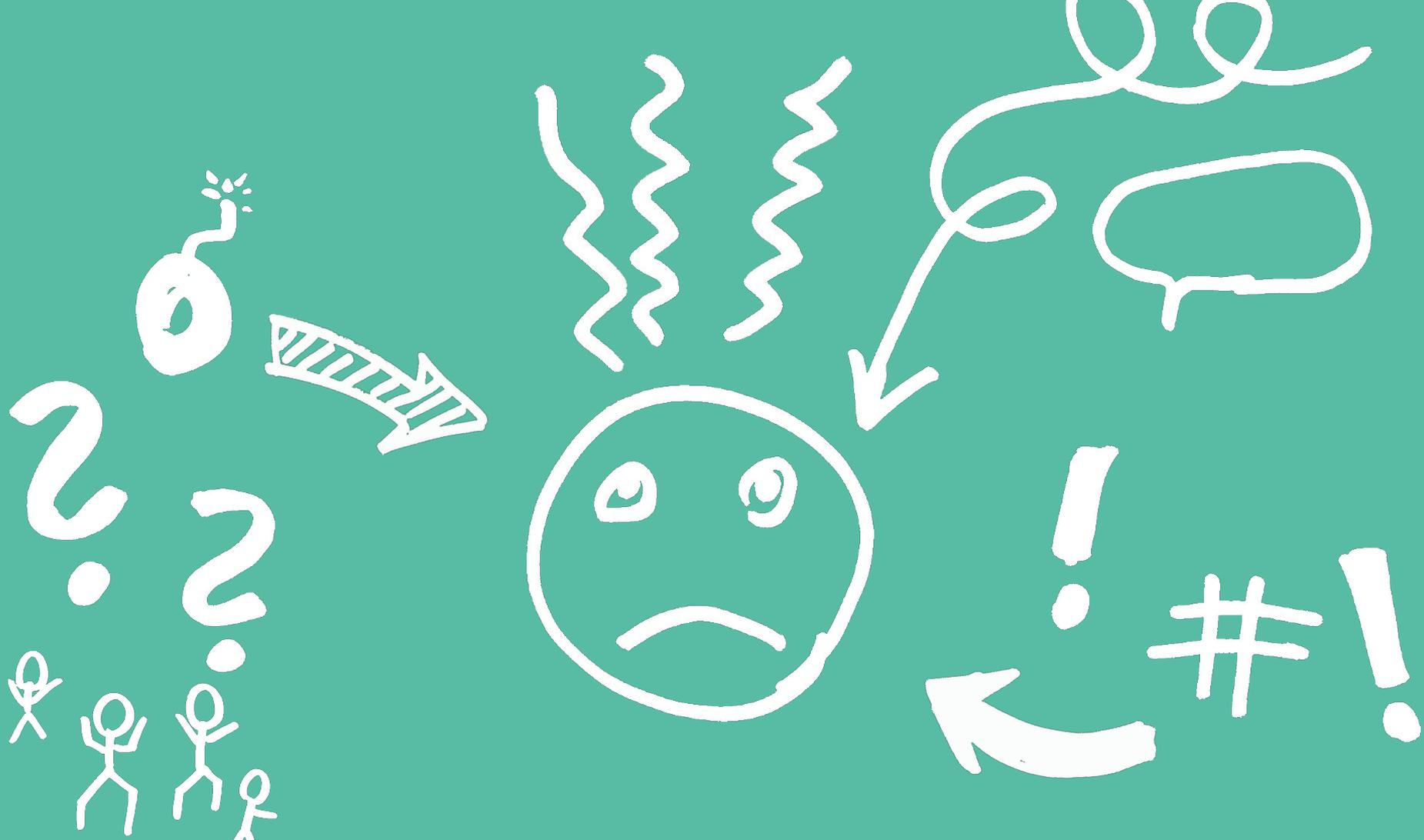
Pull Request

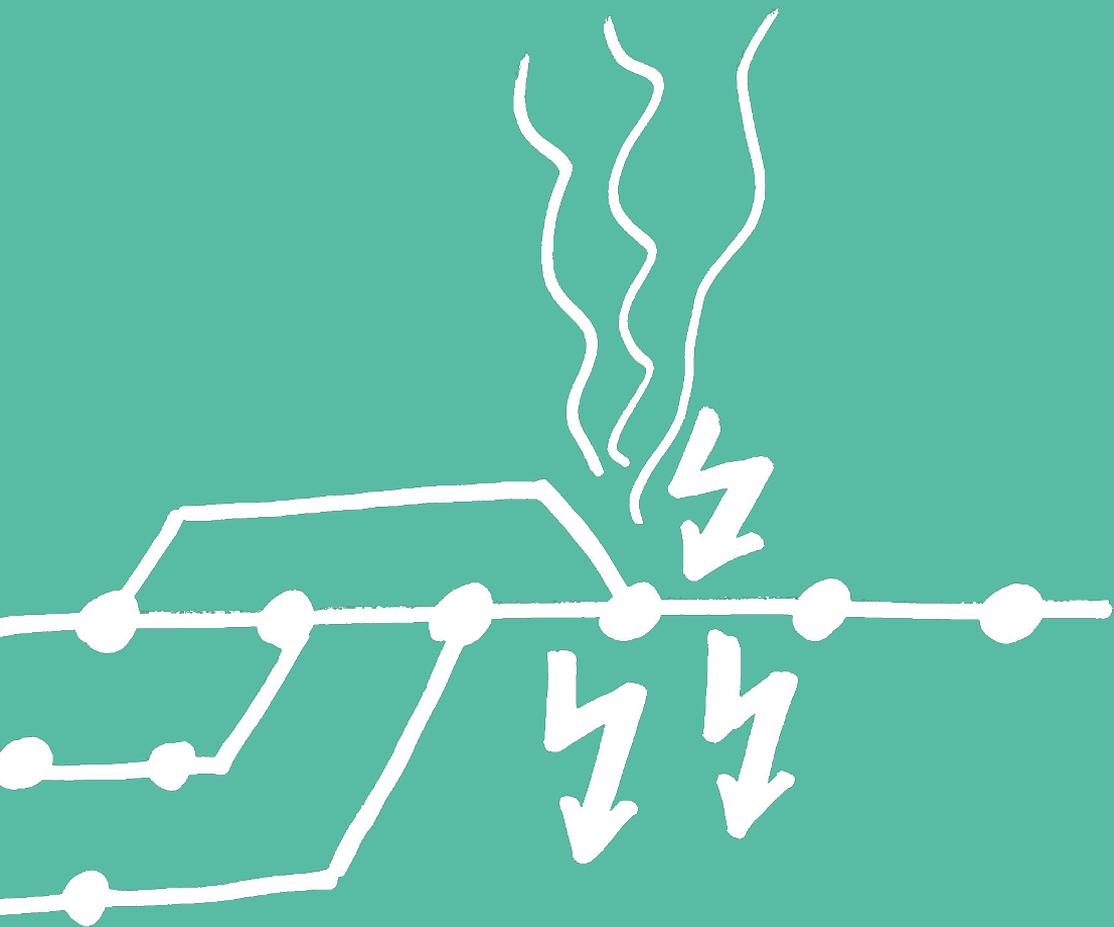


Merge

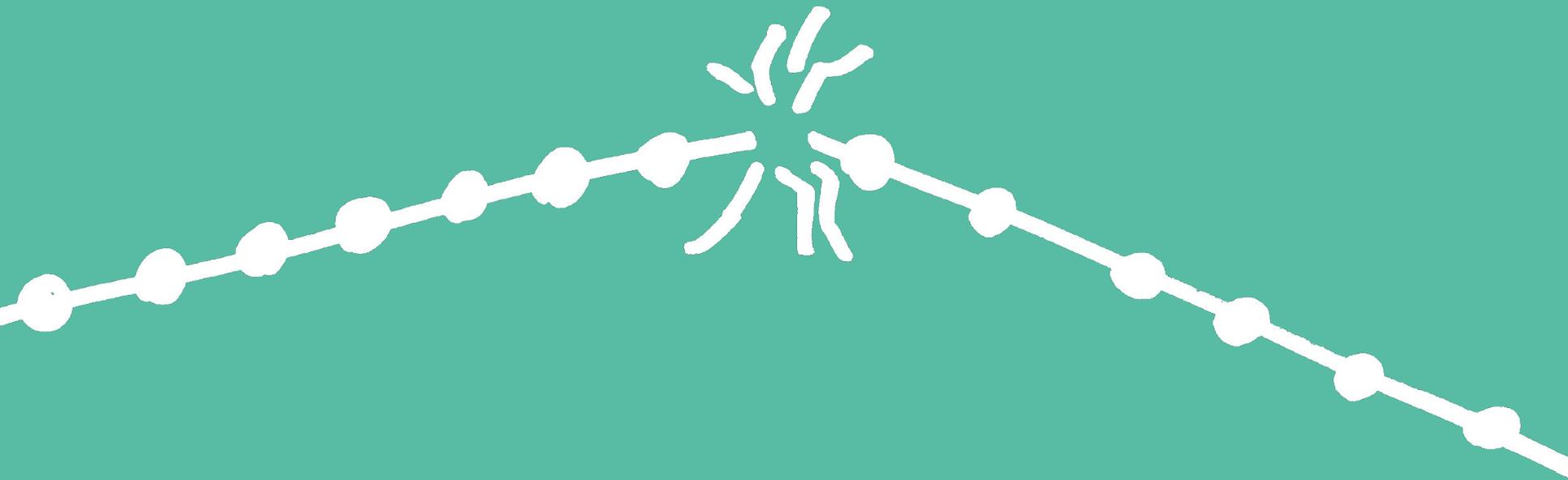






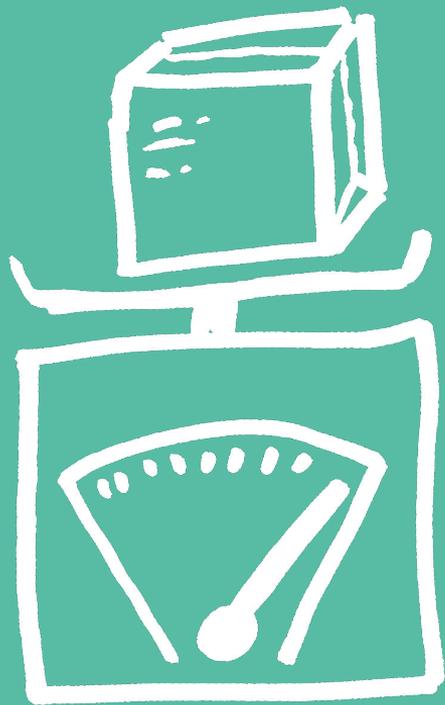






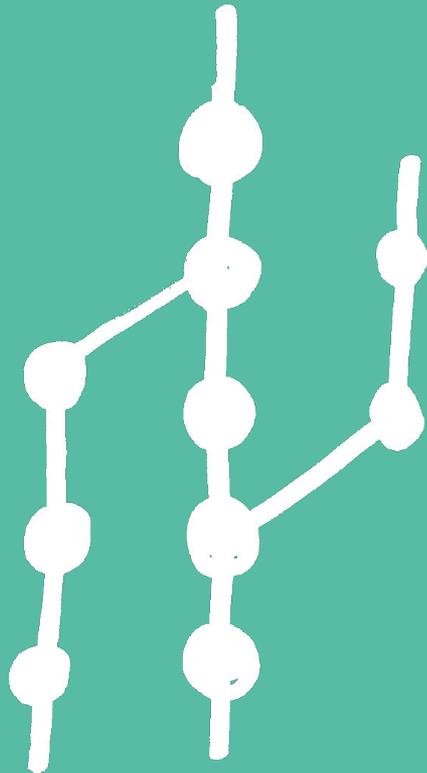


Tuesday evening

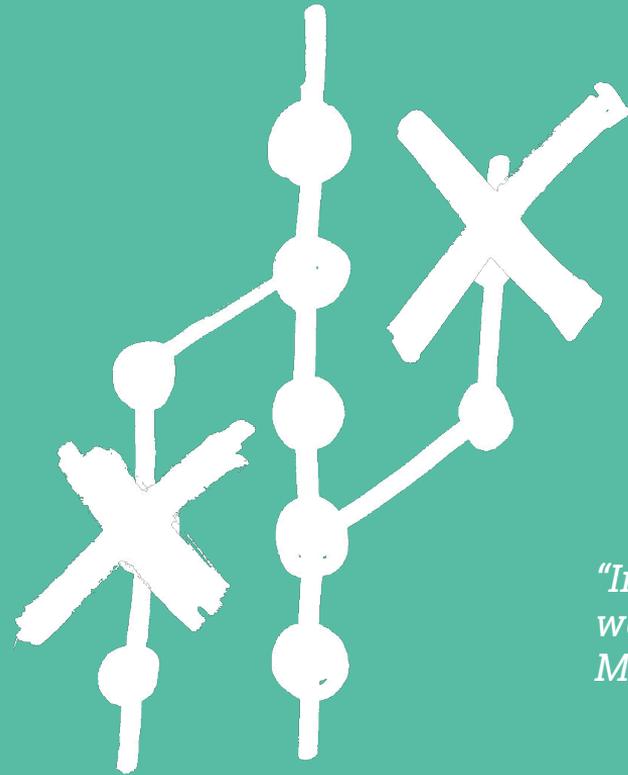


Let's do Trunk
Based Development!





“Currently, we develop features in branches. When they are ready, we merge them into master.”



*“In Trunk Based Development,
we allow just one branch: the
Master or Trunk”*



*"Everyone commits to master.
Master is being deployed to
production."*



"It's crucial that HEAD is deployable to production at any time. If the build fails, it must be fixed immediately."

"That will never work!"

"We will have tons of errors on production!"



"That will never work!"

"We will have tons of errors on production!"

We can't ensure quality like that!

Nobody will know what is released!



"That will never work!"

"We will have tons of errors on production!"

We can't ensure quality like that!

Nobody will know what is released!

Code Reviews won't work!



*Huge Commits, Broken
Head*

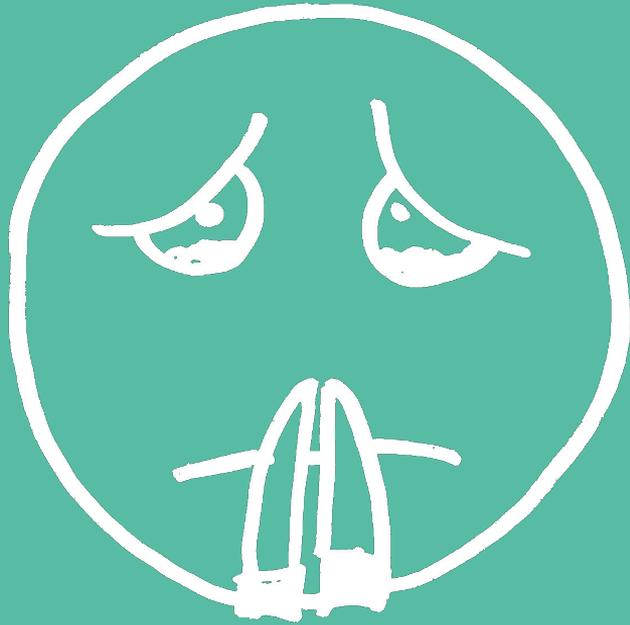


Very slow CI cycles



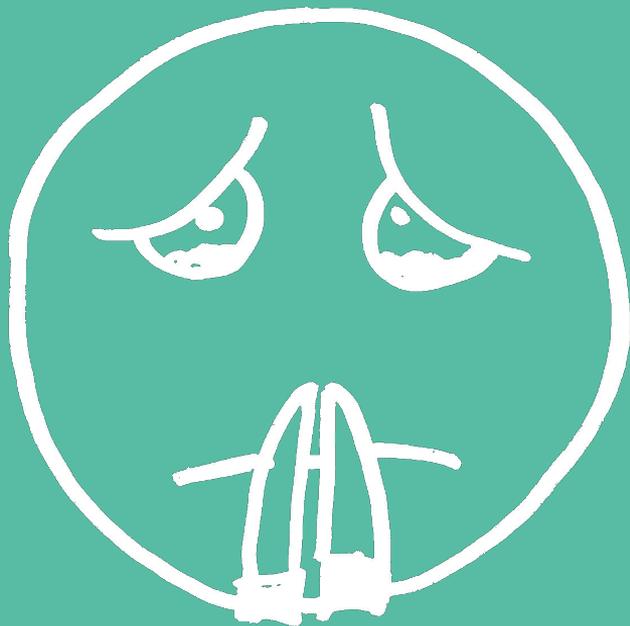
*Huge Commits, Broken
Head*





*"This feature will not
work with Trunk
Based Development"*

*“This feature will not
work with Trunk
Based Development”*



*“There is a reason, why
Github Flow exists! We put
our money on the wrong
horse!”*



*"I am paid for this,
so I just do it"*

*"Okay, we've gotta
deal with that!"*



"Okay, we've gotta deal with that!"



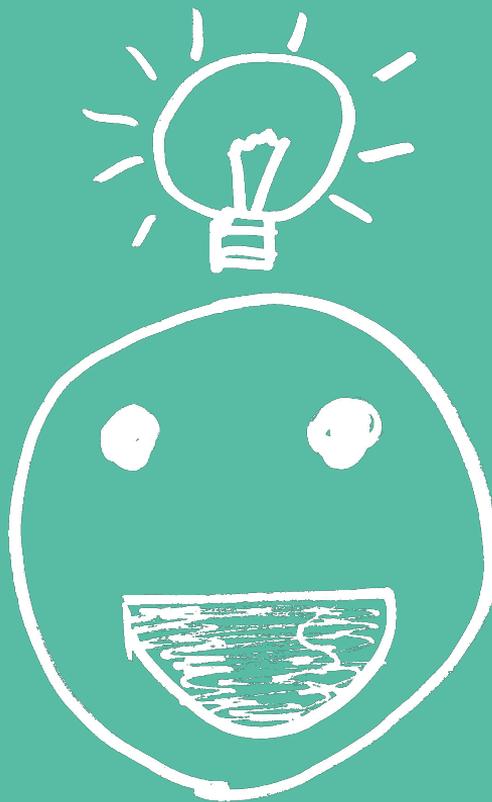
"Hm, maybe we can tune our CI to be faster, then we don't have to wait that long"

"Okay, we've gotta deal with that!"



"We should have better acceptance tests"

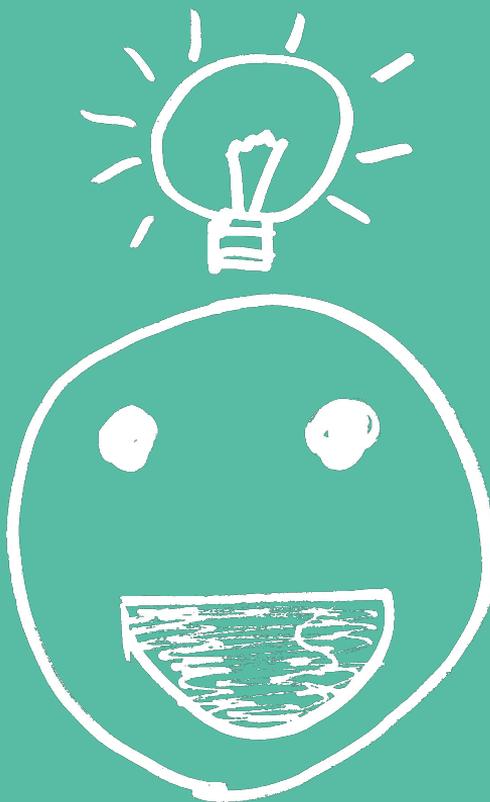
"Hm, maybe we can tune our CI to be faster, then we don't have to wait that long"



*“Actually we do
have way less
merge conflicts”*

*Our Pipeline is simpler and
faster.*

*Pushing code changes to
prod is way simpler!*



*“Actually we do
have way less
merge conflicts”*

*Our Pipeline is simpler and
faster.*

Pushing code changes to prod is way simpler!



Let's improve monitoring so that we see if a release breaks something!



"Actually we do have way less merge conflicts"

Our Pipeline is simpler and faster.

... Let's make it even better!



*When I develop Test Driven,
I can push at any time!*



*When I develop Test Driven,
I can push at any time!*



*I can build new Features
behind feature flags and
push without breaking*

*When I develop Test Driven,
I can push at any time!*



*We can break our Jira
tickets down into smaller
subparts. If I could
implement an issue in a day,
this would be perfect!*

*I can build new Features
behind feature flags and
push without breaking*

*When I develop Test Driven,
I can push at any time!*



*We can break our Jira
tickets down into smaller
subparts. If I could
implement an issue in a day,
this would be perfect!*

*I can build new Features
behind feature flags and
push without breaking*

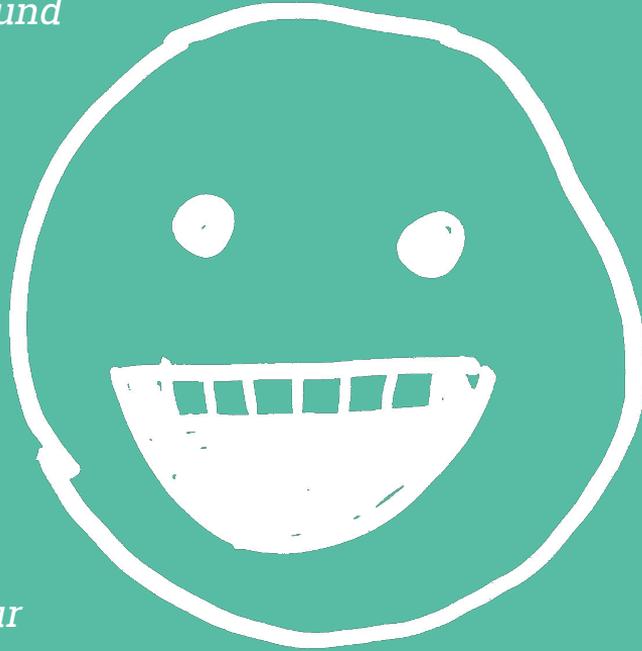
*Let's release more often! If it is
on production, we don't need to
worry about it anymore.*



"I get less interrupted by code reviews or bugs found on QA"

No rollback of hotfix in weeks

Lead Time has decreased!

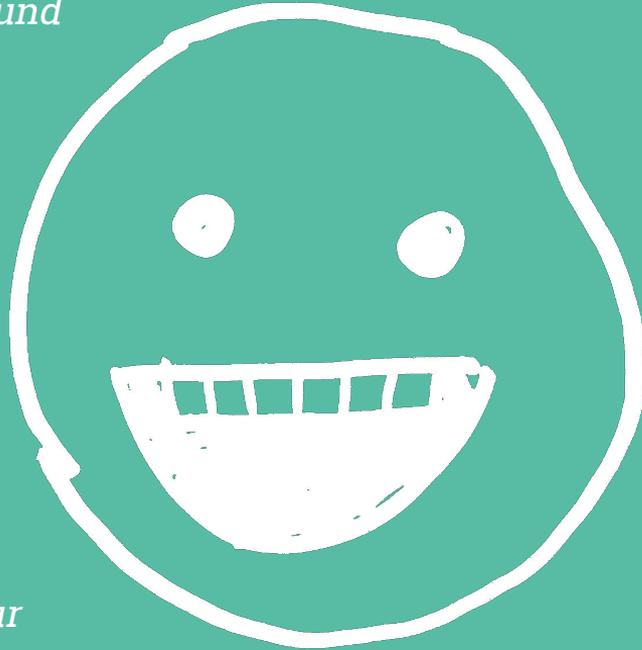


"I am very confident in our regression test suite!"

"I get less interrupted by code reviews or bugs found on QA"

No rollback of hotfix in weeks

Lead Time has decreased!



"I am very confident in our regression test suite!"

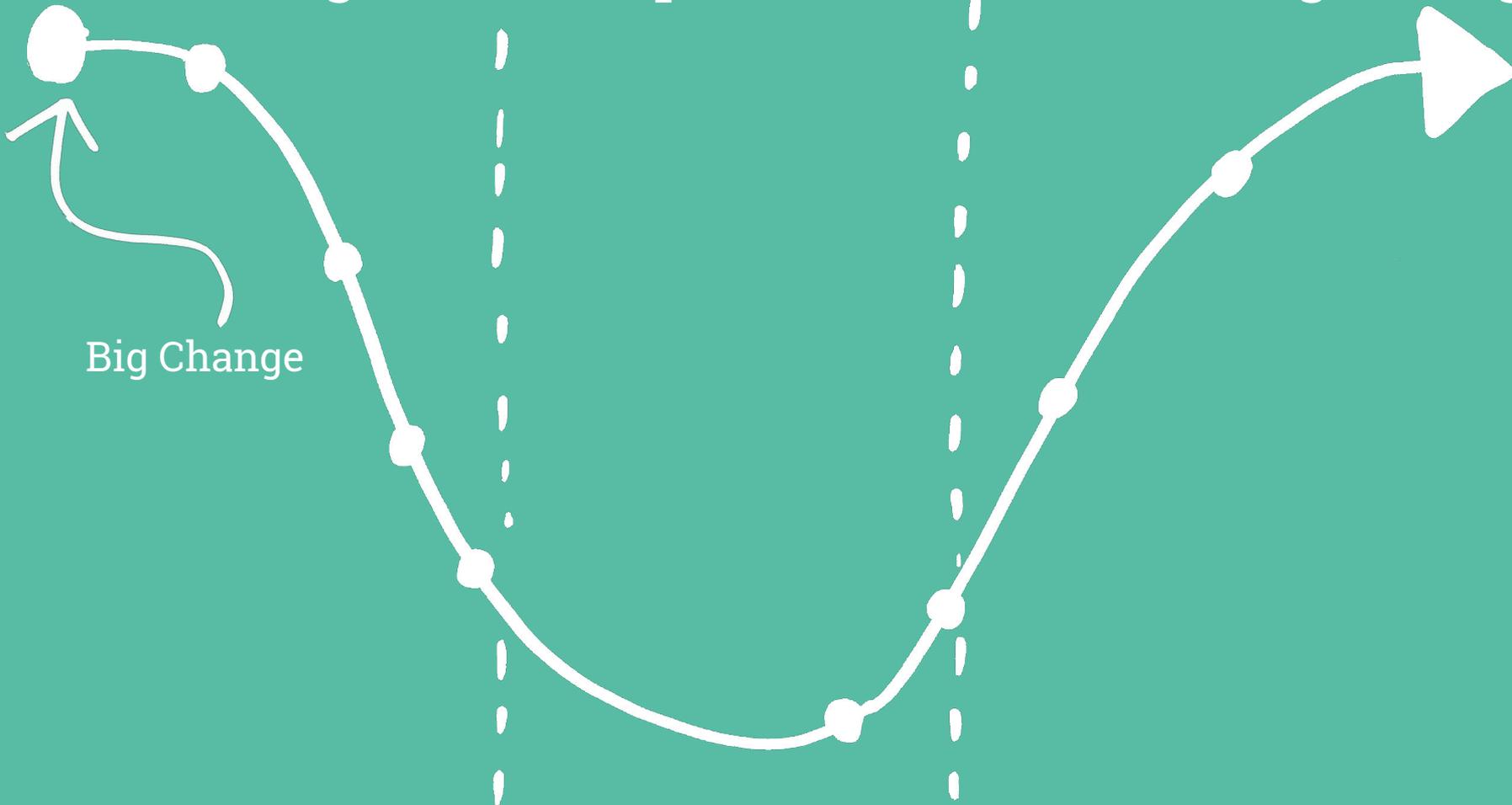
Do Pair Programming instead of Code Reviews



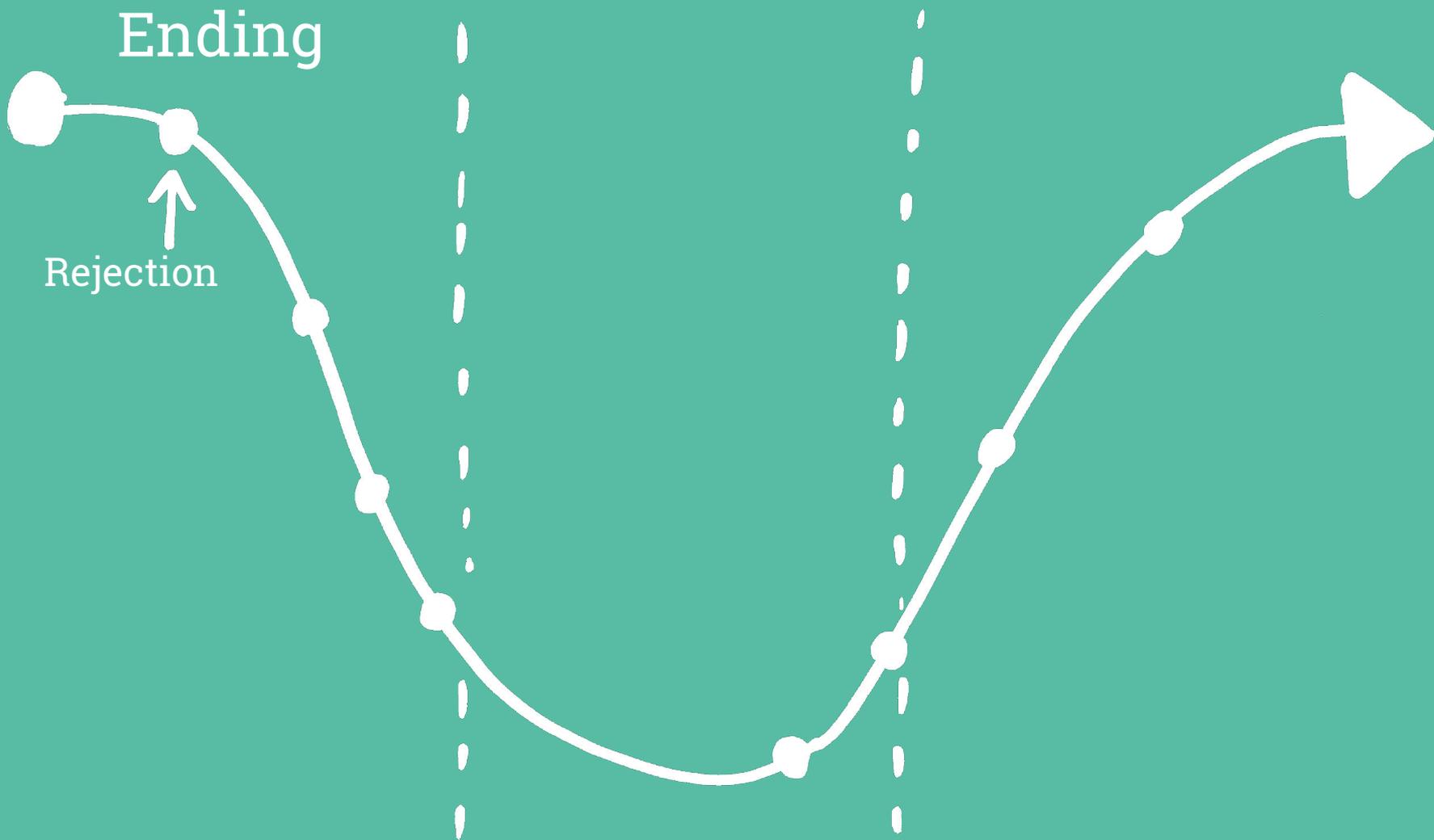
Ending

Exploration

New Beginning

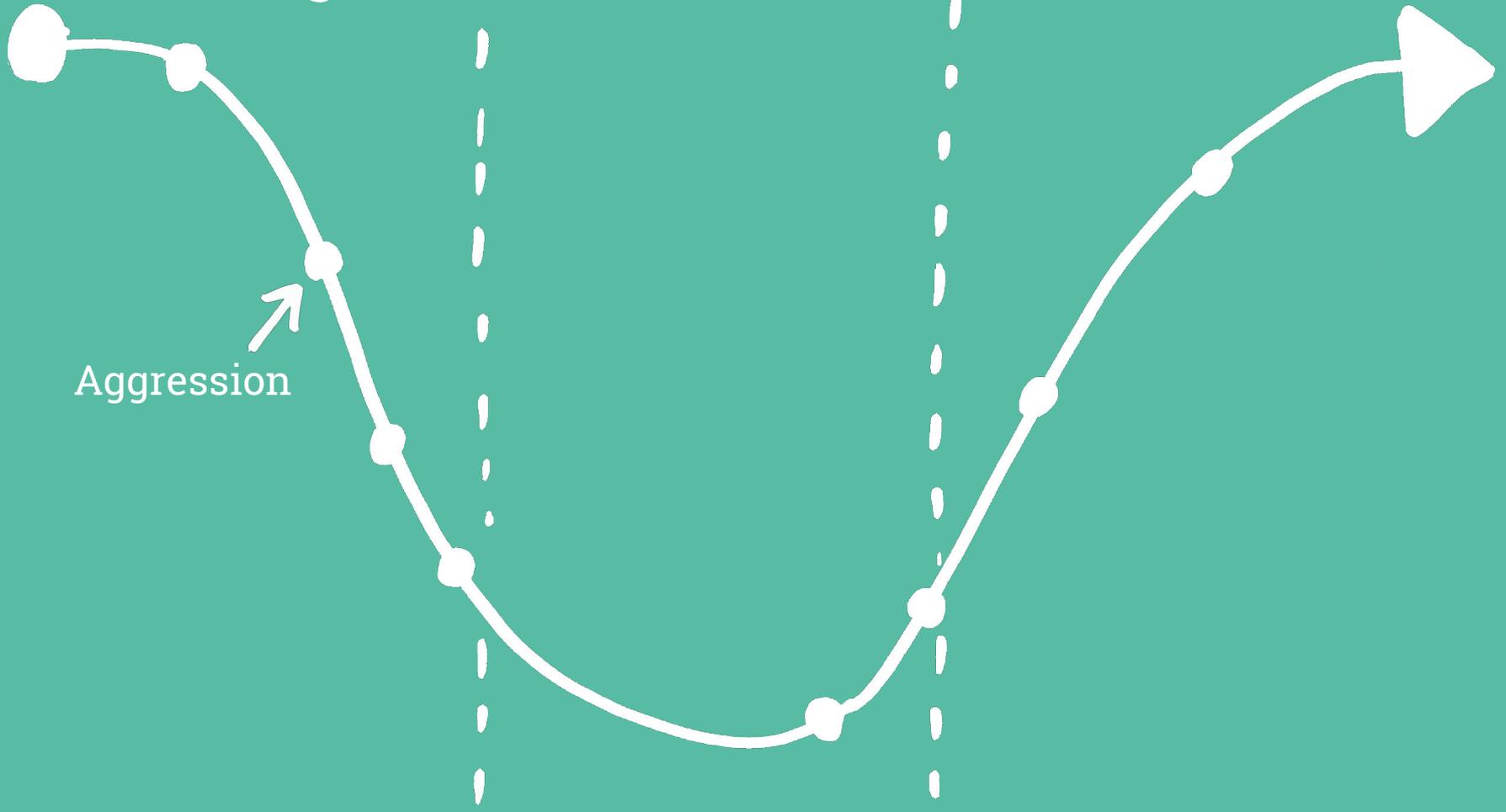


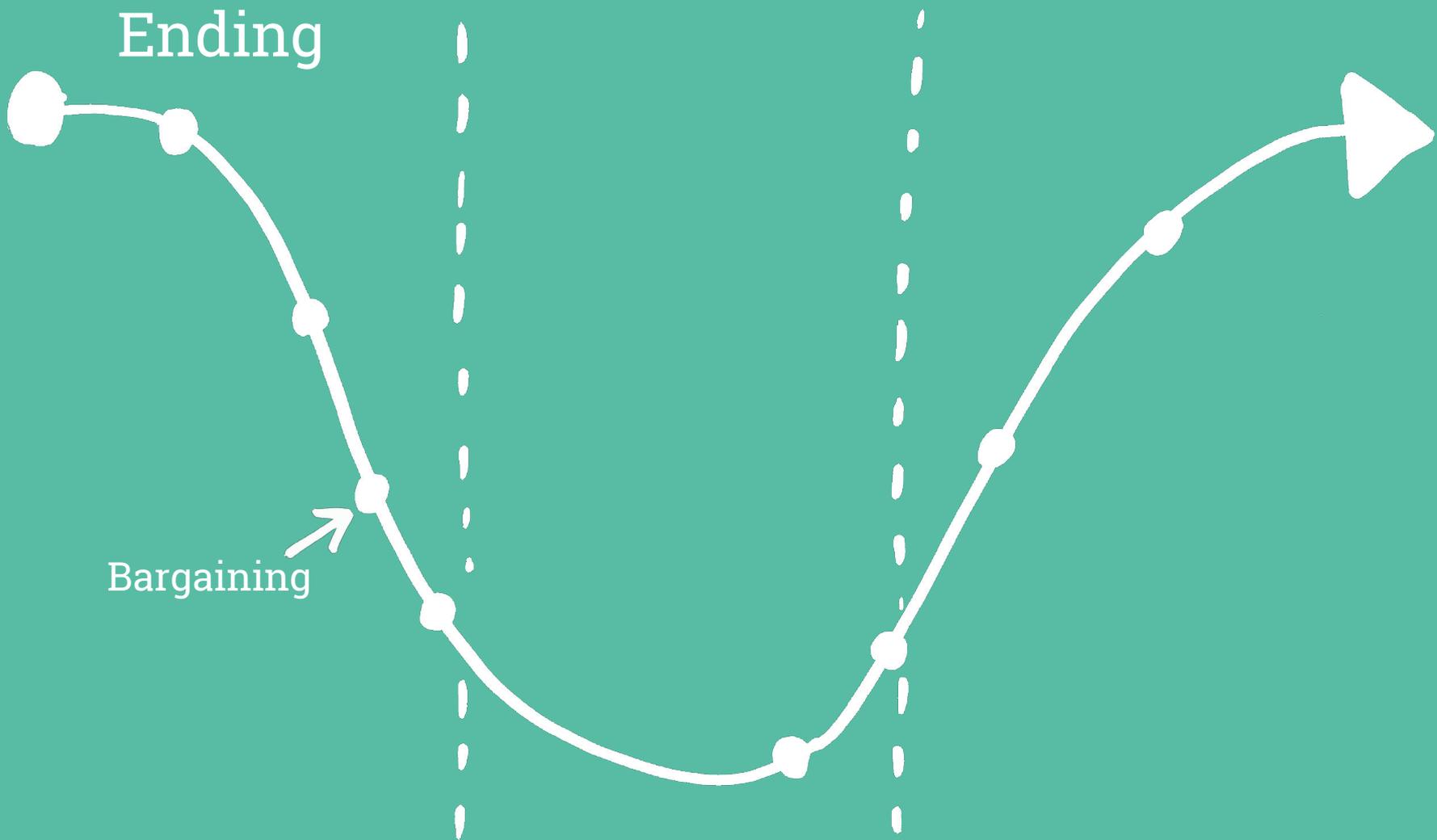
Big Change



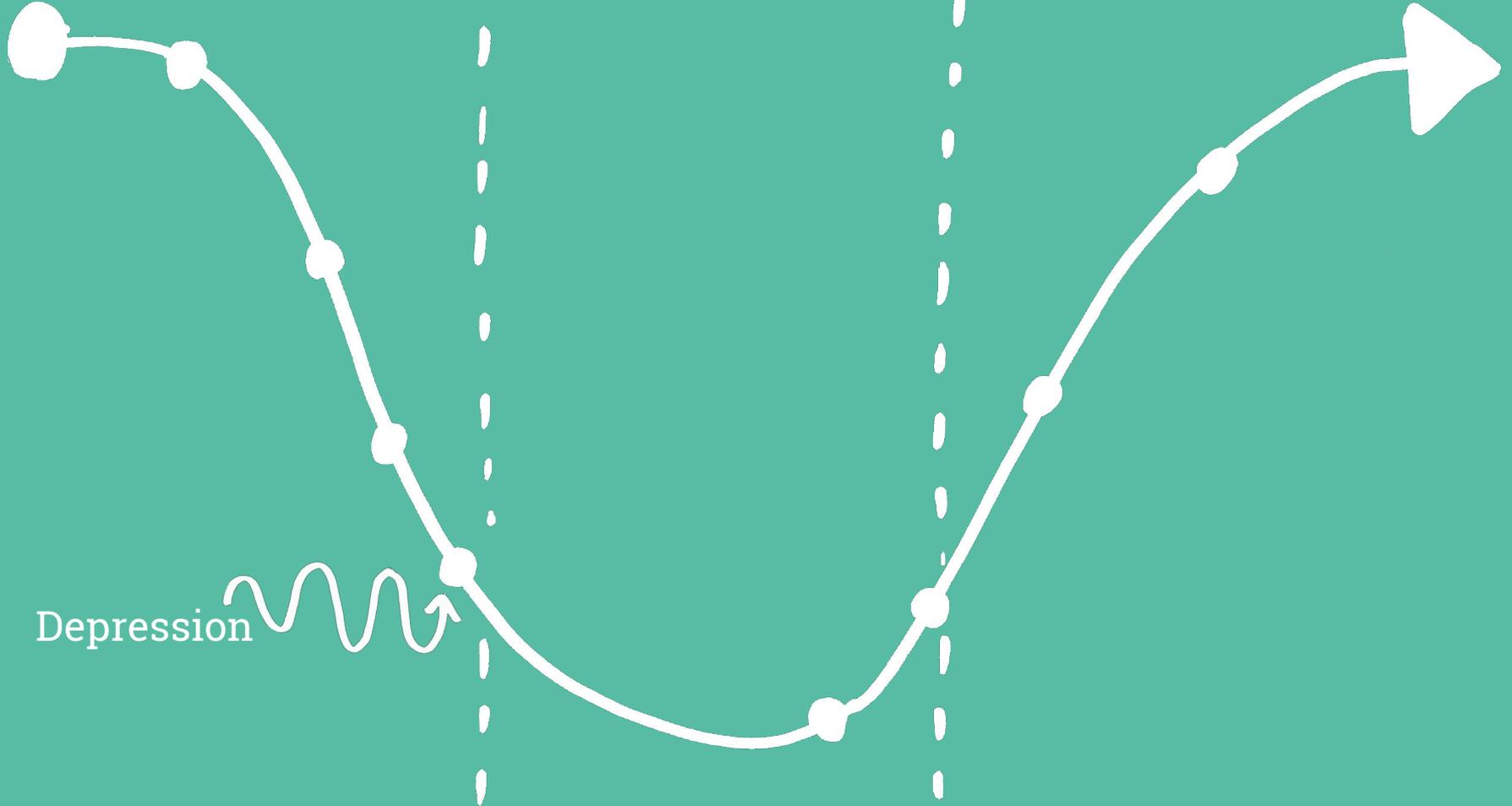
Ending

Aggression

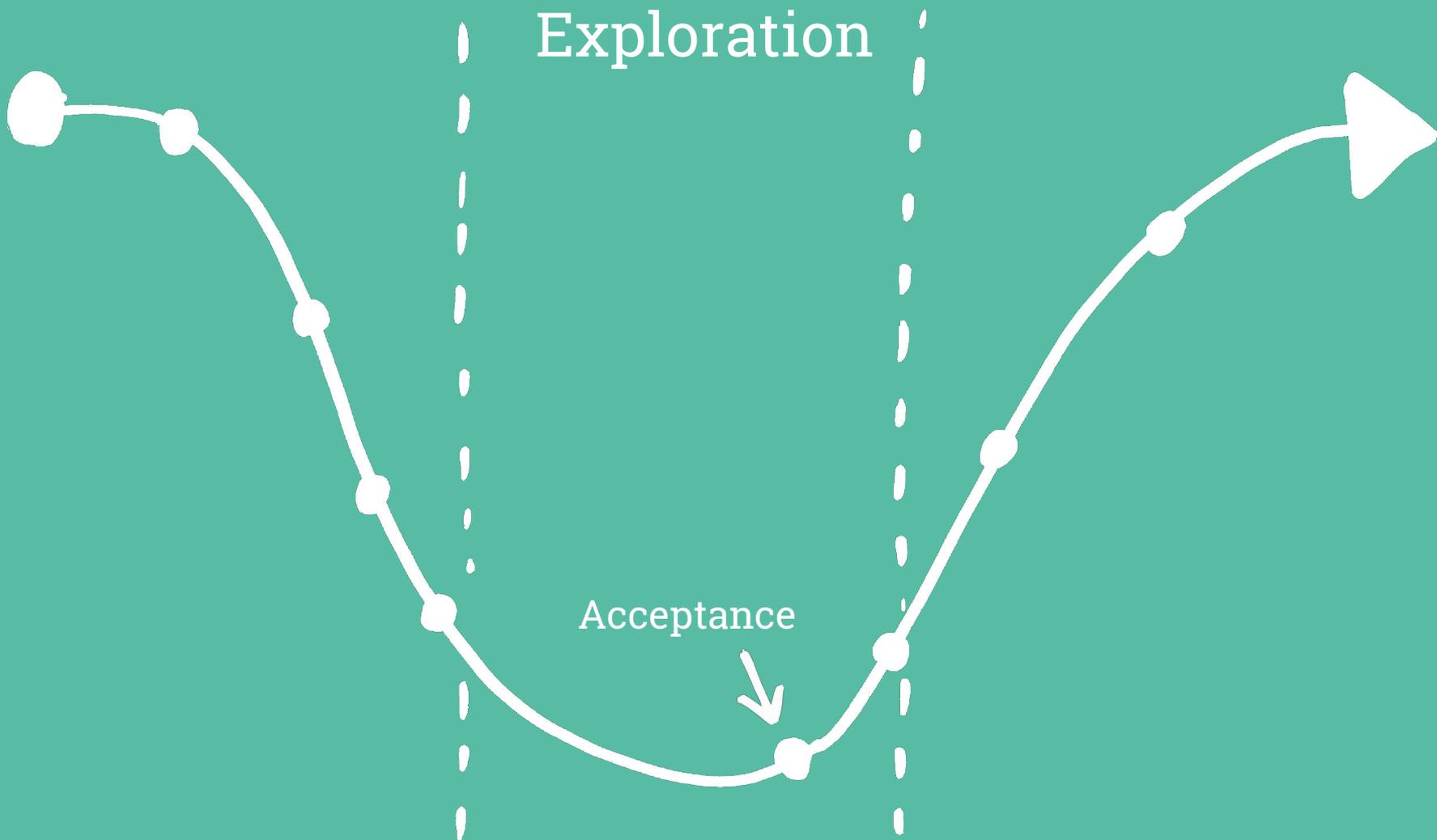


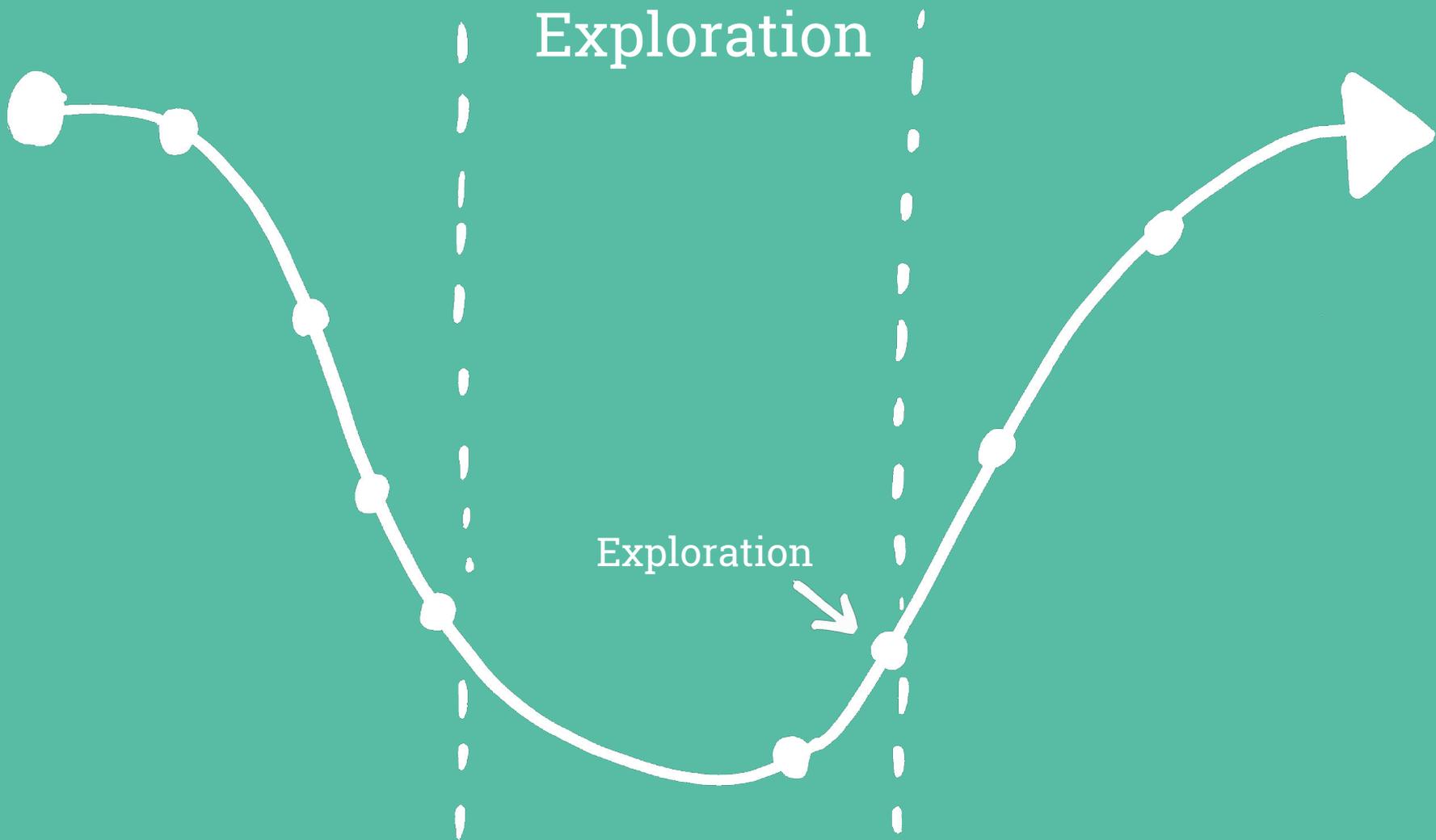


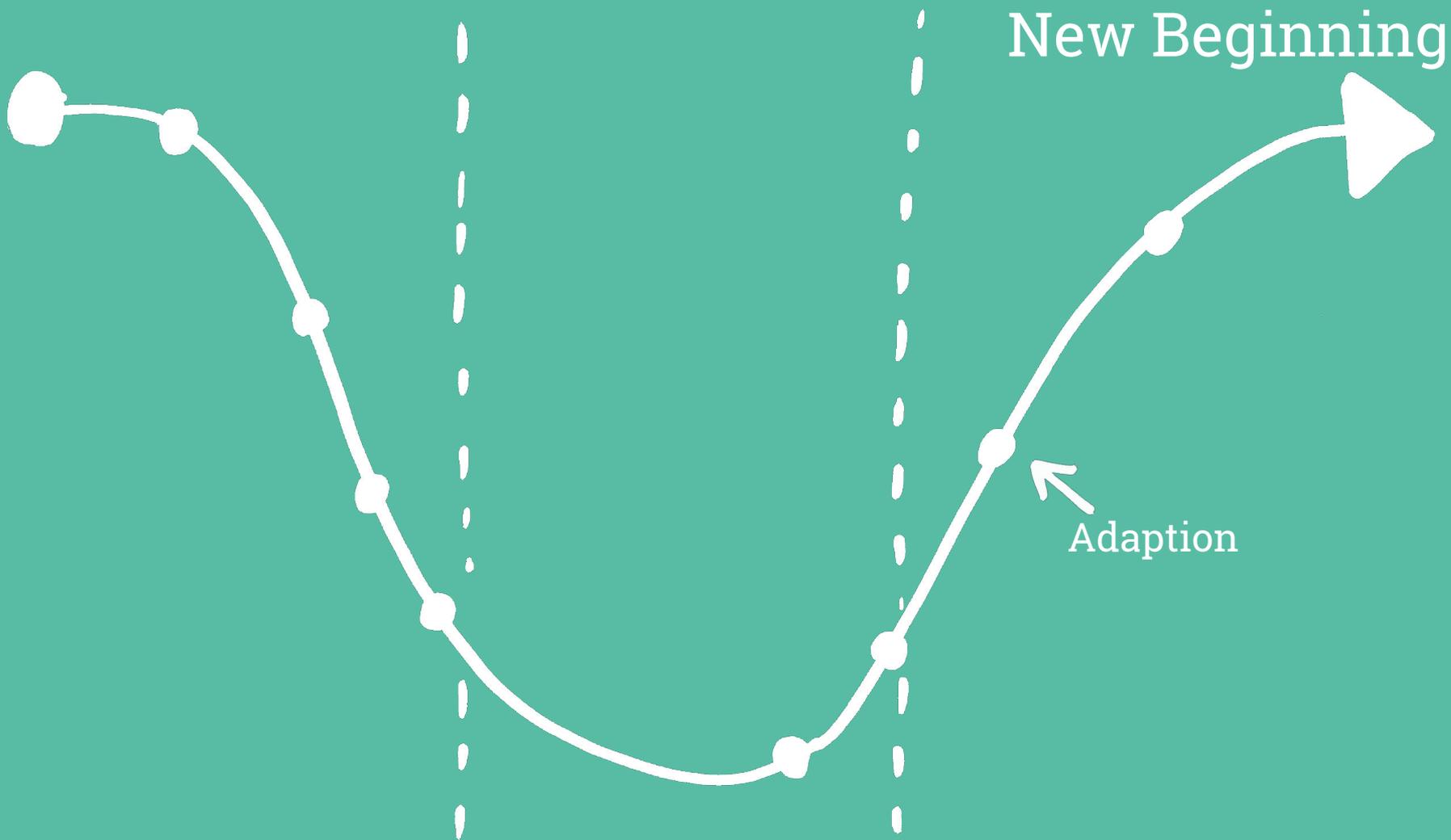
Ending

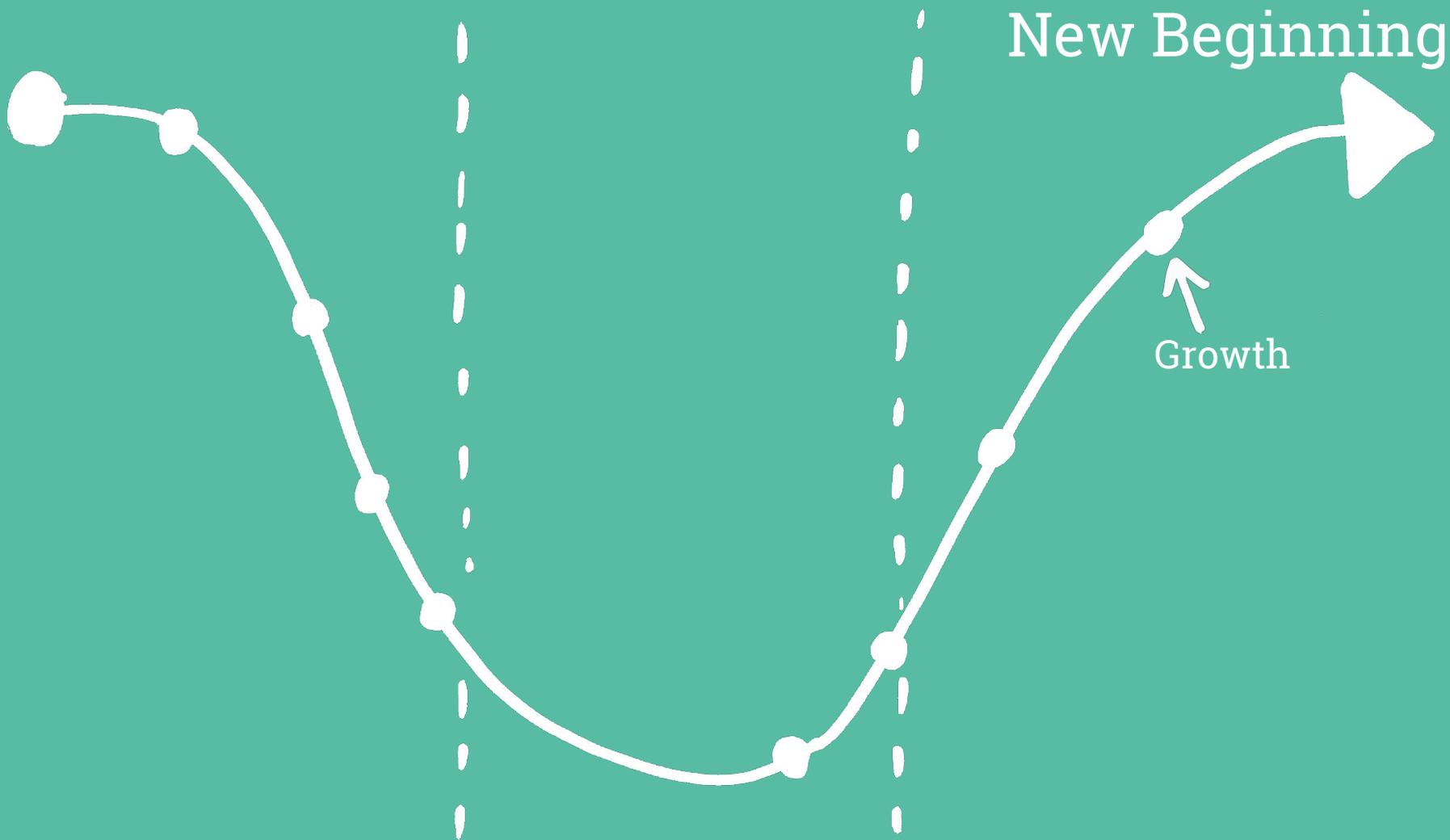


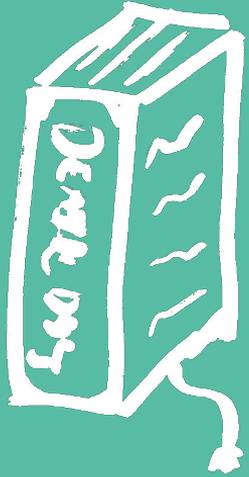
Depression



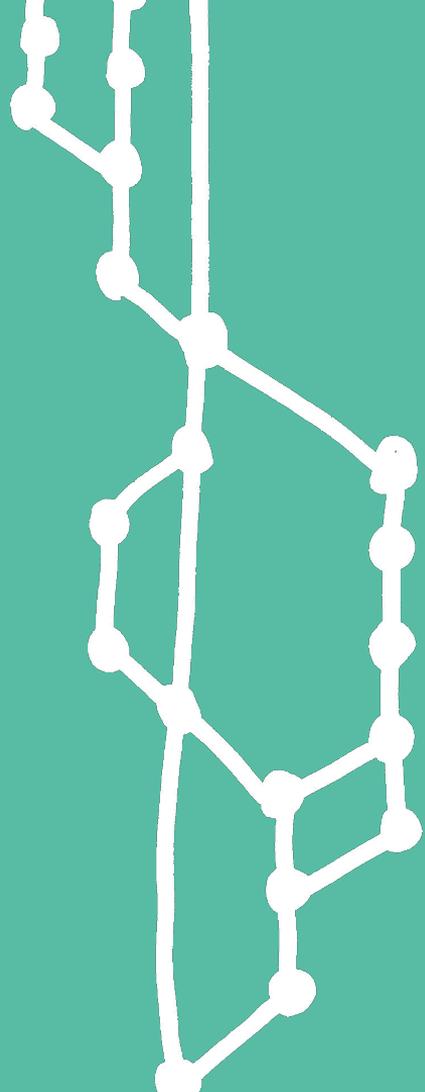




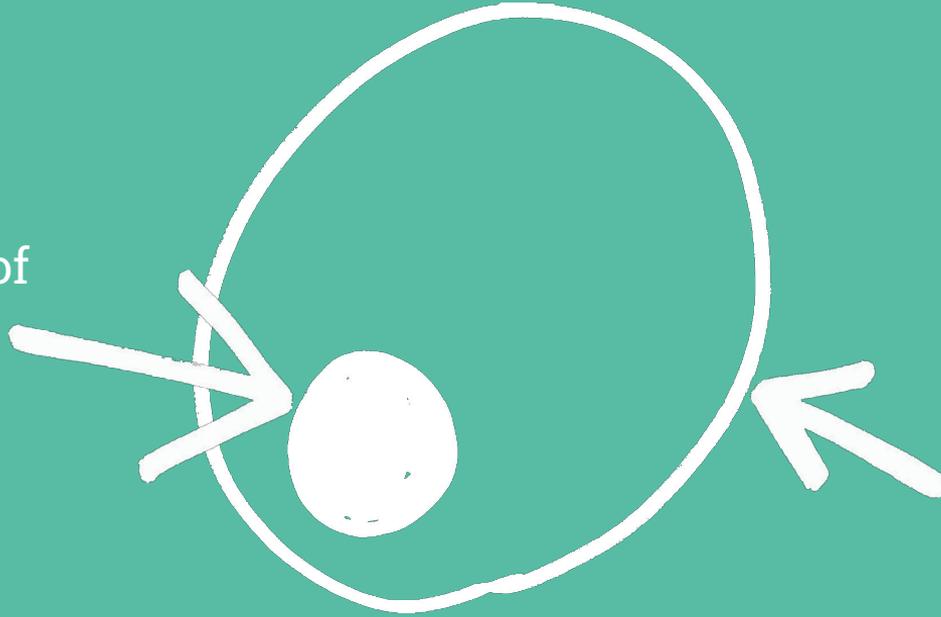




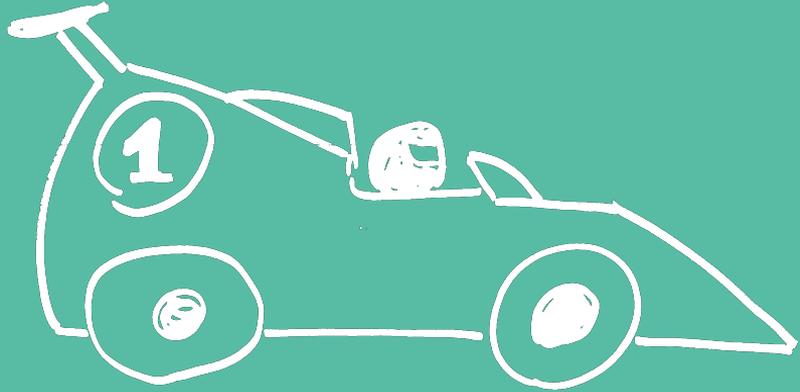
Theory: Github Flow



Circle of
Trust



Circle of
Committers



Integrate Late

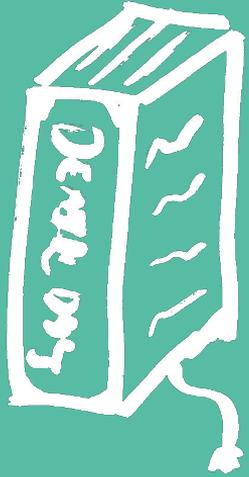
Bigger Releases - Bigger Risk

Stale Branches

Merging/Releasing is Risky and avoided

Disencourage full ownership

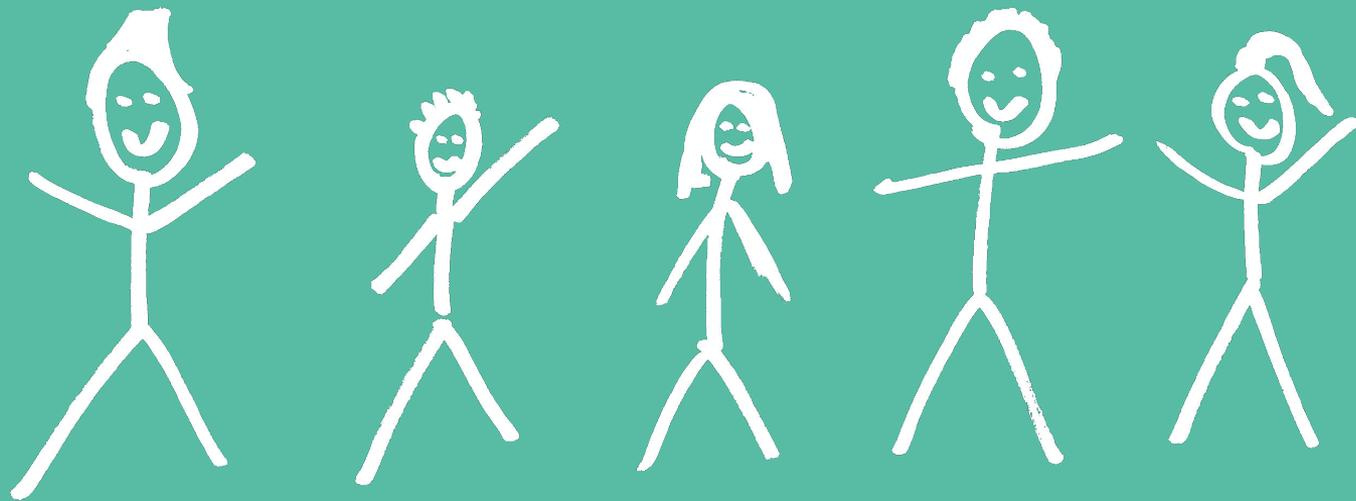


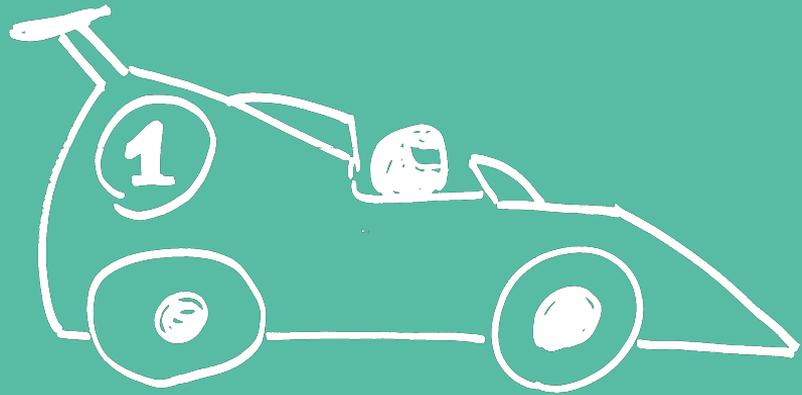


Theory:

Trunk Based Development







Messy Commit Log

Rules Change - Hard to Adapt

How do we do code reviews?

Hard for junior people and new joiners

